

14



S. GASAS

 **kelionė į  
tiesinio  
programavimo šalį**

**MATEMATIKOS MOKYKLA**

---

**S. GASAS**



**KELIONĖ  
Į TIESINIO PROGRAMAVIMO  
ŠALĮ**



**Leidykla „Mokslas“  
Vilnius — 1977**

Iš rusų kalbos vertė P. R u m š a s



G  $\frac{20204-215}{M854(10)-77}$  51-77

© Vertimas į lietuvių kalbą. Leidykla „Mokslas“, 1977

## **RUSIŠKOJO LEIDIMO PRATARMĖ**

Susidomėję kokia nors šiuolaikinės fizikos, chemijos, biologijos ar kurio kito tiksliojo mokslo problema, be vargo rasite populiarią knygą, kuri ne tik patenkintų jūsų smalsumą, bet, gal būt, ir paskatintų giliau studijuoti tą sritį. O jeigu, perskaite, sakysime, fizikos knygą, jūs iki galo nesuprasite reliatyvumo teorijos, tai vis tiek pajusite, koks įdomus tas mokslas.

Kitai yra su valdymo mokslu, ypač nagrinėjančiu įvairių visuomenės gyvenimo procesų valdymą. Nors ir nemažai šiuo metu spausdinama populiarių knygų, brošiūrų ir straipsnių valdymo klausimais, bet nėra leidinių, skirtų šiuolaikiniams matematiniams valdymo metodams. Štai kodėl, paklausę nespecialisto, kokia jo nuomonė apie matematinių metodų taikymą, pavyzdžiui, ekonominėje kibernetikoje, tikriausiai išgirstume, kad tai nepaprastai sudėtingas ir sunkiai suprantamas dalykas, be to, dar ir visai nuobodus.

Tarybiniam skaitytojui pateikiama žymaus amerikiečių mokslininko Saulo Gaso knyga išsklaidys tokią nuomonę. Moksliškai, kartu įdomiai ir gyvai autorius pasakoja apie matematinius metodus visuomenės gyvenimui tirti ir konkrečiai apie tiesinio programavimo metodus bei jų praktinį pritaikymą.

Kokia gi tiesinio programavimo metodų esmė? Pameginsime trumpai atsakyti į šį klausimą šiek tiek bendriau, negu tai daroma „Kelionėje į Tiesinio Programavimo Šalį“.

Įvairiausiose mūsų gyvenimo ir veiklos srityse nuolat pasitaiko iš esmės vienokių uždavinių: žinodami (visiškai arba iš dalies) esamą situaciją ir galimus alternatyvius tolesnės veiklos variantus; turime nuspręsti, kurį variantą pasirinkti, o kurių atsisakyti. Tai, galima sakyti, ir yra valdymo problema. Ji gali būti susijusi tiek su atski-

žmogaus arba žmonių grupės veikla, tiek su vienu ar kitu ūkiniu procesu arba visos ekonomikos vystymu, tiek su kokių nors operacijų paruošimu ir pan. Vadinas, būtina pasirinkti vieną iš leistinų (galimų) variantų, t. y. sudaryti planą. Sveikas protas pataria pasirinkti geriausią, naudingiausią variantą. Tačiau pats klausimas „Kokį variantą pasirinkti?“ nėra per plauką nepriartina prie valdymo uždavinio sprendimo. Jis to uždavinio net neapibrėžia, nes visiškai neaišku, ką reiškia žodžiai „geriausias variantas“, „naudingiausias variantas“. Tą išsiaiškinti ir yra viena svarbiausių problemų, kurios neišsprendus, neįmanoma nei suformuluoti, nei išspręsti valdymo uždavinio. Iš esmės čia kalbame apie valdomos sistemos išvystymo tikslą. Tikslai gali būti labai įvairūs: nagrinėjant ūkinį objektą, galima turėti tikslą pagaminti maksimalų produkcijos kiekį, kai turimi resursai yra riboti, sunaudoti gamybai minimalų kokios nors žaliavos kiekį, gauti maksimalias pajamas ir pan. Nurodžius tikslą, sąvoka „geriausias“ savaime pasidaro griežtai apibrėžta. Geriausias, arba, kaip dažniausiai sakoma, optimalus, variantas tenkina dvi sąlygas: pirma, jis yra vienas iš leistinų variantų, antra, veikiant pagal tą variantą, pasiekiamas iškeltojo tikslo maksimumas arba minimumas (žiūrint prasmės).

Vadinas, bendroji valdymo uždavinių prasmė yra gana paprasta, nors patys uždaviniai kartais būna labai sudėtingi ir ne visada išsprendžiami šiuo metu žinomais metodais. Šiais samprotavimais buvo pagrįsta daugybė valdomojo matematinio programavimo srities tyrimų.

Valdymo uždavinys matematinio programavimo priemonėmis formuluojamas šitaip. Formaliai užrašomas valdomojo objekto veiklos sąlygų kompleksas, vadinamas apribojimų sistema. Pavyzdžiui, tiriant įmonę, nurodomi resursai, kuriuos ji gali suvartoti (nebūtinai visus), taip pat gamybinių procesų, kuriais tie resursai paverčiami gaminiiais, galimybės. Ieškomi uždavinio kintamieji — gamybinių procesų intensyvumai. Pagal fizinę prasmę jie yra neneigiami dydžiai. Jeigu intensyvumas lygus nuliui, tai procesas nevyksta.

Vadinas, valdomojo objekto veiklos sąlygų kompleksas išreiškiamas lygčių ir nelygybių sistema, iš kurios gaunama tolesnio jo vystymo leistinų variantų aibė.

Optimalus variantas parenkamas, naudojantis vadinamąją tikslo funkcija. Pavyzdžiui, žinodami savo gaminių

kainas, stengiamės, kad, realizavę produkciją, gautume maksimalias pajamas.

Sudarius apribojimų sistemą ir tikslo funkciją, valdymo uždavinys jau yra suformuluotas. Tada reikia ieškoti jo optimalaus sprendinio — nustatyti, kokius gamybinius procesus reikia vykdyti ir koks turi būti kiekvieno proceso intensyvumas, kad, nepereikvojus įmonei skirtų resursų, būtų gautos maksimalios pajamos. Išsprendus uždavinį, galima sudaryti optimalų tos įmonės vystymo planą.

Kaip spręsti tokį uždavinį? Paprasčiausias metodas, atrodytų, peršasi savaime. Pakanka, numčius visus leistinus variantus, apskaičiuoti, kiek pajamų duotų kiekvienas, ir savaime paaškėtų, kuris variantas geriausias. Tačiau iš tikrųjų tas metodas yra sudėtingiausias, o dažniausiai tiesiog nerealus, nes dauguma matematinio programavimo uždavinių turi tiek leistinų variantų, kad praktiškai neįmanoma jų aprėpti arba jų skaičius net yra begalinis. Kadangi atsakymo rasti tiesioginiu būdu (arba, kaip sako matematikai, analiziškai) dažniausiai negalima, tai išnagrinėti variantus — ne tokia jau absurdiška idėja. Ja pagrįsti daugelis algoritmų; vienas iš jų — simplekso metodas, kurį aprašo S. Gasas. Sprendžiant tais metodais, tiriami ne visi variantai, o tik maža jų dalis.

Tokia yra bendra matematinio programavimo uždavinių prasmė. Jeigu matematinio programavimo apribojimų sistema ir tikslo funkcija yra tiesinės, t. y. užrašytosios lygtys ir nelygybės yra pirmojo laipsnio, tai sprendžiamas tiesinio programavimo uždavinys. Tiesinis programavimas — geriausiai teoriškai ištirta matematinio programavimo sritis, turinti milžinišką taikomąją reikšmę, nes didesnė dalis konkrečių programavimo uždavinių, kurie praktiškai sprendžiami, naudojantis ESM,—tai tiesinio programavimo uždaviniai.

Tiesinis programavimas gimė mūsų šalyje. Žymaus tarybinio matematiko ir ekonomisto, Lenino premijos laureato, akademiko L. Kantorovičiaus darbe „Matematiniai gamybos organizavimo ir planavimo metodai“, išspausdintame 1939 m., pirmą kartą buvo suformuluotas tiesinio programavimo uždavinys, aprašantis realią ekonominę situaciją, ir rastas jo sprendimo algoritmas. Be to, Kantorovičius pirmasis pasiūlė optimizuoti viso liaudies ūkio planavimą, ir iš tos idėjos susiformavo ištisa mokslinio tyrimo kryptis. Taigi „Kelionės“ autorius klysta, teigda-

mas, kad tiesinis programavimas atsirado penktojo dešimtmečio pabaigoje. Tuo metu tikslai prasidėjo intensyvus tiesinio programavimo vystymasis, susijęs su operacijų tyrimu JAV. Operacijų tyrimo tikslas — išnagrinėti, kaip reikia organizuoti valdomojo objekto veiklą, kad ji būtų efektyviausia, ir teikti atitinkamus nurodymus objekto vadovams. Tuos klausimus dažniausiai sprendžia ekspertų grupė, į kurią įeina matematikas, sudarantis objekto modelį arba, kai objektas per daug sudėtingas, kai kurių jo dalių modelius. Operacijų tyrimas JAV prasidėjo per antrąjį pasaulinį karą, sprendžiant karinius klausimus. Karinės (o ne ūkinės, kaip mūsų šalyje) problemos buvo akstinas JAV išvystyti ir tiesinį programavimą. Net tokie, atrodytų, nekalti uždaviniai, kuriuose nagrinėjama, kaip sudaryti optimalų riešutų mišinį arba kaip pagaminti pigiausią pašarą galvijams, atsirado, planuojant karines operacijas. Taikiems tikslams šį mokslą imta taikyti gerokai vėliau, kai paaiškėjo, jog tiesinis programavimas gali duoti didelį ekonominį efektą, sprendžiant visokius ūkinius klausimus. Operacijų tyrimų grupės tada virto specializuotomis konsultacinėmis firmomis, teikiančiomis paslaugas įmonėms. F. Burlackis teigia, kad JAV „dabar yra 150 konsultacinių firmų, sudarinėjančių konkrečias programas firmoms, concernams, įmonėms. Tos konsultacinės organizacijos parduoda savo paslaugas Vakarų Europos ir kitų pasaulio šalių klientams. Apie 20 tūkstančių svarbiausių JAV kompanijų ir šimtai stambiausių kitų šalių firmų bei concernų naudojami konsultacinių firmų paslaugomis“\*.

Konkrečių uždavinių, sprendžiamų tiesinio programavimo metodais, ratas per paskutinius metus nepaprastai išsiplėtė; apie tai labai įdomiai pasakoja S. Gasas.

Vis dėlto autorius nenušviečia kelių svarbių tiesinio programavimo klausimų. Aptardamas uždavinių sprendimo metodus, S. Gasas nenurodo jų ekonominės interpretacijos. Tuo tarpu optimalaus sprendinio ieškojimo procesas turi gana aiškią ekonominę prasmę. Ją turime suvokti toli gražu ne vien iš tuščio smalsumo: žinodami algoritmo ekonominį turinį, geriau suprasime, kaip mąsto planuotojas, nesinaudojantis matematiniais metodais ir ESM, ku-

---

\* Ф. Бурлацкий. Надежды и иллюзии. — «Новый мир», 1972, № 7, с. 154.

rie jo samprotavimai gali būti klaidingi ir kaip išvengti klaidų tais atvejais, kai dėl vienos ar kitos priežasties neįmanoma sudaryti objekto matematinio modelio.

Knygoje tik užsimenama apie vadinamąjį dualų uždavinį, nors iš šio uždavinio kintamųjų savybių paaiškėjo, kad tai — svarbus tyrimo įrankis. Kainų sudarymui, naudojantis matematinio programavimo metodais, šiandien skiriamas ypatingas dėmesys.

Pagaliau, knygoje nėra nė žodžio apie tokią svarbią problemą, kaip liaudies ūkio planavimo ir valdymo optimizavimas visos šalies mastu. Ši problema, aišku, netelpa tiesinio programavimo rėmuose, tačiau, ją tyrinėjant, kyla daug panašių klausimų, kaip ir ieškant ekonominio objekto vystymo optimalaus plano. To priežastys aiškos ir susijusios su JAV visuomeninės santvarkos pobūdžiu.

Vadinasi, nors beveik visi S. Gaso knygoje pateikti pavyzdžiai yra iš ekonomikos, pačios knygos negalima laikyti ekonomine. Vis dėlto lokalinių ekonominių procesų modeliavimo, naudojantis tiesinio programavimo metodais, taip pat optimalių sprendinių ieškojimo pagrindai išnagrinėti labai rimtai. Reikia tik turėti omenyje, kad daugelis konkrečių uždavinių yra būdingi JAV socialinei ekonominei santvarkai, todėl jų negalima mechaniškai taikyti mūsų sąlygoms. Tačiau objekto matematinio modelio sudarymo principas, matematinio uždavinio tipas ir jo sprendimo metodai yra tokie patys.

Šiai knygai skaityti nereikia specialių matematinių žinių, todėl skaitytojų ratas gali būti labai platus. Labiausiai norėtusi rekomenduoti šią knygą jaunimui, ieškančiam, kur pritaikyti savo kūrybines jėgas. Ji labai pravers inžinieriams, ekonomistams, planuotojams ir visiems, kas nori susipažinti su vienu iš svarbiausių dabartinės valdymo teorijos metodų.

Tikiuosi, kad keliionė po Tiesinio Programavimo Šalį iš tikrųjų bus įdomi ir, be abejo, padės „iškirsti langą“ į sudėtingą ir patrauklų vienos šiuolaikinio mokslo šakos pasaulį.

Baigiant norėtusi priminti vieno iš žymiausių pasaulio matematikų Leonardo Oilerio žodžius: „Viskas, kas vyksta pasaulyje, turi kokį nors maksimumą arba minimumą“. Todėl galimybės pažinti pasaulį optimizavimo metodais iš tiesų yra neribotos.

*J. Ovsijenka*

Mokymo menas — tai menas žadinti jaunose sielose žinių troškimą, o pažadinus — patenkinti jį; sveiką, gyvą smalsumą turi lydėti gera nuotaika, o jei žinios į galvą kemšamos prievarta, jos tik slegia ir užteršia protą. Norint suvirškinti žinias, reikia jas ryti su apetitu.

*Anatolis Fransas*

## PRATARMĖ

*...daiktų pradai nepasiekiami akiai...*

Valdymo mokslas ir su juo susijusių techninių priemonių bei metodų arsenalas traukia gana plačios ir vis gausėjančios auditorijos dėmesį. Kadangi į galingą ir daugiašakį valdymo mokslą atveria duris tiesinis programavimas, tai aš tikiuosi, kad elementari knyga apie jį tai auditorijai galėtų būti naudinga daugeliu aspektų. Vieniems ji, gal būt, tarsi katalizatorius nuties tiltą į specialiąją literatūrą, o kiti pasitenkins ja, susipažinę su tuo, kas juos domina. Šiaip ar taip, noriu tikėtis, kad viena ar dvi valandos, praleistos „kelionėje“ (ir kai kurie apmąstymai apie joje pavartotą, beje, labai ribotą, matematinį aparatą), padės skaitytojui suvokti pačius esmingiausius tiesinio programavimo bruožus. Vadinas, „Kelionės į Tiesinio Programavimo Šalį“ tikslas — elementariai, bet gana rimtai supažindinti (kartu ir sudominti) skaitytoją su vienos šiuolaikinio mokslo srities pagrindinėmis sąvokomis.

Kadangi aš aktyviai dirbu tiesinio programavimo srityje nuo 1952 metų, tai man dabar sunku iš daugelio faktorių griežtai išskirti tuos, kurie lėmė galutinį šios knygos variantą. Jos užuomazgą galima rasti 1954 metų nespacialaus pobūdžio publikacijoje. Visą tolesnį laiką mano domėjimasi šia tema palaikė ir skatino knygos, paskaitos, susitikimai, konferencijos ir, be abejo, pokalbiai su draugais ir kolegomis. Su dėkingumu pripažįstu didžiulį jų indėlį. Noriu pareikšti gilią padėką ir Bilui Makviljamui už puikias iliustracijas, taip gerai atspindinčias knygos esmę.

*Saulas I. Gasas*

## 1. ĮVADAS

(Apie tai, kaip greičiausiai apsirengti, apie tieses ir apie pirmuosius žingsnius Tiesinio Programavimo Šalyje)

Nuo labai senų laikų žmonės pradėjo ieškoti geriausio sprendinio (maksimumo, minimumo ir apskritai sprendinio, optimalaus vienu ar kitu atžvilgiu). Jau Euklidas aprašė, kokiais būdais galima nubrėžti ilgiausią ir trumpiausią atkarpą, jungiančią duotąjį tašką su apskritimu, parodė, kaip iš žinomo perimetro lygiagretainių rasti tą,



kurio plotas didžiausias. Didieji XVII ir XVIII amžiaus matematikai išvystė naujus optimizavimo metodus, kuriais galima spręsti ištisą geometrijos, mechanikos ir fizikos uždavinių kompleksą.† Prie tokių uždavinių priskiriamas, pavyzdžiui, minimalių sukimosi paviršių ir sparčiausio nusileidimo kreivės nustatymas.

Mūsų laikais atsirado visiškai naujos klasės uždavinių, susijusių su sudėtinga organizacine struktūra, labai

dažnai pasitaikančia šiuolaikinėje visuomenėje. (Mūsų natūralus polinkis kelti ir spręsti tokius uždavinius išreiškiamas posakiais „kuo mažiausiai sąnaudų“, „kuo naudingiau“, „kuo pelningiau“ ir pan.) Jiems priklauso tiek klausimai, kaip efektyviausiai valdyti kokį nors ūkį ar įmonę arba optimaliausiai išvystyti kokią nors pramonės šaką, tiek ir visai „žemiški“ uždaviniai, sakysime, kaip sudaryti pigiausią ir racionaliausią pašarą galvijams. Mėginant tiksliai suformuluoti ir išspręsti tokius uždavinius, teko sukurti naujus, labai svarbius optimizavimo metodus. Vienas tų metodų — tiesinis programavimas — ir aptariamas šioje knygoje. Nevartojant specialių terminų, jį galima nusakyti dviem būdais: paaiškinant paraidžiui žodžių „tiesinis programavimas“ reikšmę, arba tiesiog išvardijant tipinius uždavinius, sprendžiamus tuo metodu. Kiekvienas tų būdų turi savo pranašumų, todėl pasinaudosime abiem.

Programavimas, tiesinis ar kitoks, plačiausia to žodžio prasme yra susijęs su efektyviausio ribotų išteklių panaudojimo arba paskirstymo uždaviniais. Tokie uždaviniai ekonomikoje yra svarbiausi. Tačiau jie iškyla ne tik pramonėje ir ūkyje, bet ir kiekvieno žmogaus kasdieniniame gyvenime, pasireikšdami įvairiausiomis formomis.

### **———— Aple tai, kaip greičiausiai apsirengti**

*Ak, tas Džonas mažulėlis  
Ruošės gult su kojinelėm,  
Bet, batuko nenutraukęs,  
Guli lovoj geltonplaukis!*

Pirmasis uždavinys, kurį sprendžiame kiekvieną rytą, — apsirengimo programavimo uždavinys. Turime sudaryti veiksmų programą, pagal kurią galėtume apsirengti, atsižvelgdami į tam tikrus apribojimus arba į visų pripažintas taisykles (kojines apsimaui būtina, bet ne ant batų ir t. t.). Laikas — mūsų pagrindinis išteklius, todėl pasirinktoji programa turi būti geriausia ta prasme, kad geriausiai, kaip kam atrodo, būtų išnaudotas rytinis laikas.

Aš pats, atmetęs „neesmines detales“, į savo programą įtraukiu veiksmus su šešiais daiktais: batais, kojinelėmis, kelnėmis, marškiniais, kaklaraiščiu ir švarku. Veiksmų programa nurodo, kuria eile tuos daiktus reikia apsivilkti ar apsiauti. Šiuo atveju galima sudaryti net 6×5×

$\times 4 \times 3 \times 2 \times 1 = 720$  skirtingų programų. Kai kurios programos netinkamos, nes arba netenkina pripažintų apribojimų (kojinės ant batų), arba nepraktiškos (kaklaraištis po marškiniais). Tačiau, net atsisakius tų netinkamų sprendinių, vis tiek reikės ištirti tam tikrą leistinių programų skaičių.



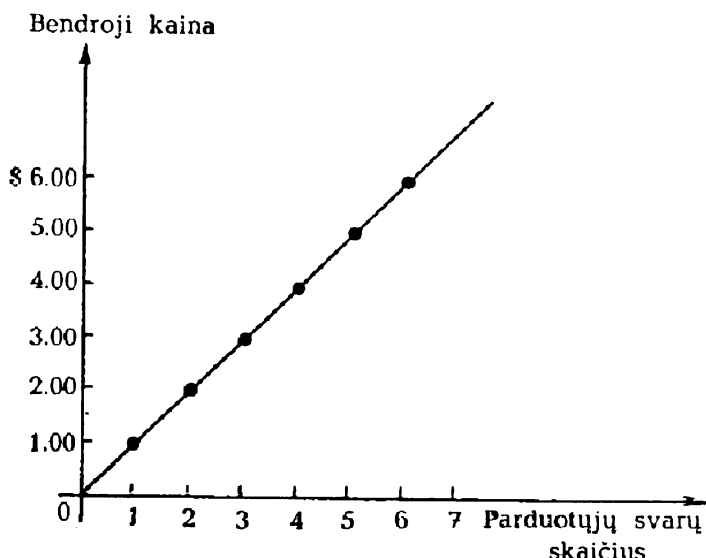
Kaip pasirinkti galutinį, optimaliausią sprendinį? Kiekvieno uždavinio sprendinys vertinamas tam tikru efektyvumo matu, pagrindiniu tikslu, pagal kurį galima palyginti leistinių programų efektyvumą. O jeigu aš galiu koku nors būdu palyginti kelių programų matus, tai iš tų programų galiu pasirinkti optimaliausią. Spręsdamas pateiktąjį konkretų uždavinį, noriu minimizuoti laiką, sugaištamą apsirengti. Tai — mano efektyvumo matas (programavimo kalba — tikslo funkcija), kuriuo remdamasis galiu palyginti įvairius leistinius sprendinius. Tiesą sakant, aš nesu išsprendęs to uždavinio sekundės tiks-

lumu, bet jau daug metų optimalia laikau tokią eilę: kojinės, marškiniai, kelnės, kaklaraištis, batai, švarkas. Tai — mano optimaliausias sprendinys; jis minimizuoja laiką, sugaištamą apsirengti, nepažeisdamas visų pripažintų apribojimų. Kas nors kitas, gal būt, remiasi kita tikslo funkcija, pavyzdžiui, siekia kuo mažiau kartų atidaryti ir uždaryti spintos dureles bei stalčius, t. y. minimizuoti rytinį triukšmą. Todėl jo optimalusis sprendinys gali skirtis nuo manojo.

Nors aptartasis uždavinys ir nėra tiesinio programavimo, bet jis yra būdingas programavimui, nes turi daug leistinių sprendinių. Jeigu būtų tik vienas leistinas sprendinys, tai apskritai nebūtų jokios problemos ir to sprendinio ieškoti nebūtų įdomu. Spręsdami kai kuriuos uždavinius, iš visų leistinių sprendinių galime pasirinkti bent

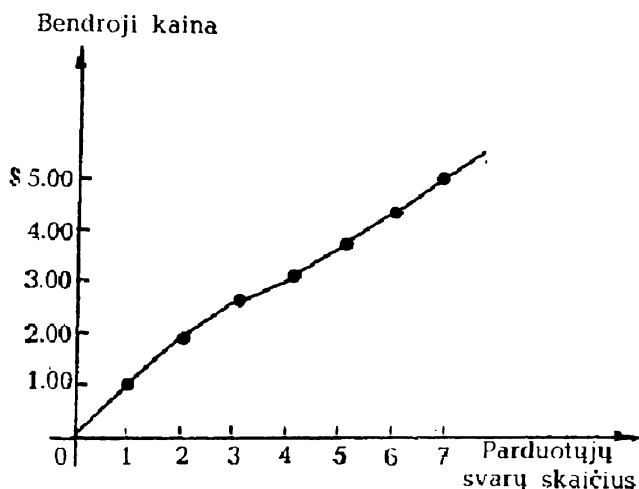
vieną optimalų sprendinį. Kaip rasti tuos leistinus sprendinius ir kaip iš jų pasirinkti optimalų, aiškinsime vėliau.

Kadangi kalbėsime tik apie tiesinio programavimo uždavinius, būtina pabrėžti, kad jie sudaro tam tikrą bendrojo programavimo, dažniausiai vadinamo matematiniu, uždavinių poaibį. Tiesinio programavimo uždaviniai nuo



kitų programavimo uždavinių skiriasi tuo, kad jie išreiškiami tiesinėmis priklausomybėmis (priklausomybėmis, kurias vaizduojame tiesėmis). Pavyzdžiui, jeigu vienas svaras kavos kainuoja 1 dolerį, o perkant didesnę prekių kiekį pardavėjas nedaro nuolaidos, tai bendroji pirkinio kaina yra proporcinga jo masei, išreikštai svarais. Kitaip sakant, jų priklausomybę galima išreikšti tiese, kurią matome šiame puslapyje pateiktame paveiksle. Priešingai, kai pardavėjas nuleidžia 10 centų nuo antrojo parduodamo svaro kainos, 20 centų — nuo trečiojo ir t. t. iki penktojo svaro ir pagaliau po 50 centų — nuo kiekvieno tolesnio svaro, kainos linija, pavaizduota 14 puslapyje pateiktame paveiksle, nėra tiesė. Taigi tiesinio programavimo uždaviniai yra tie bendrojo (matematinio) programavimo uždaviniai, kuriuose priklausomybės (įvairūs apribojimai ir tikslo funkcija) yra tiesinės. Nors ši sąlyga gali atrodyti pernelyg griežta, vis dėlto ji galioja daugeliu svarbių atvejų ir gerokai palengvina sprendinio ieškojimą.

Programavimo, ypač tiesinio, uždaviniai iškyla įvairiausiomis situacijomis. Būdingesnėms iš jų yra sudaryti standartiniai programavimo metodų taikymai, pavyzdžiui: žemės ūkyje, naftos bei popieriaus pramonėje, transporte, produkcijos paskirstyme bei inventoriaus kontrolėje, karo dalykuose, technikoje, ekonomikoje ir t. t. (žr. bibliogra-



fiją knygos pabaigoje). Išnagrinėsime kai kuriuos taikymus ir jų pavyzdžiais paaiškinsime pagrindines tiesinio programavimo sąvokas. Tačiau prieš tai reikia šiek tiek nukrypti į šalį ir išsiaiškinti, kokią vietą užima tiesinis programavimas bendrajame šiuolaikinio valdymo, arba operacijų tyrimo, moksle.

## Operacijų tyrimas ir modeliai

*Jei užsigeidžiame ką nors statyti,  
Pirmiausia planą dėstome, paskui  
Užmanymus ant popieriaus nubraižom.  
O namo kontūrus lape išvydę,  
Jau apie kainą pradedam galvoti.*

Sąvokos „operacijų tyrimas“ (OT) apibrėžimai, kaip ir patarimai įsimylėjusiems, visuomet turi paklausą, todėl jų yra daugiau, negu reikia. Apibrėžimas gali būti platus, detalizuotas, sakysime, šitoks: „operacijų tyrimas yra

mokslinių metodų ir priemonių taikymas kurios nors sistemos valdymo uždaviniams spręsti, leidžiantis rasti tų uždavinių optimalius sprendinius“, bet gali būti ir trumpas, pavyzdžiui, šitoks: „operacijų tyrimas — tai kiekybiškas sveikas protas“ arba „tai valdymo tyrimas“. Mums OT reiškia mokslinį valdymo uždavinių sprendimą. Tačiau labiausiai domina ne formalus OT apibrėžimas, o jo metodologija.

Bet kurio projekto, pagrįsto OT, sudarymą galima apytiksliai suskirstyti į šešias fazes, neturinčias griežtų ribų ir iš dalies viena kitą dengiančias\*. Dažniausiai nurodomos šios fazės:

- uždavinio formulavimas;
- tiriamosios sistemos matematinio modelio sudarymas;
- sprendinio ieškojimas, naudojantis tuo modeliu;
- modelio ir sprendinio tikrinimas;
- sprendinio tikslinimas;
- sprendinio taikymas praktikoje.

(Čia įvedėme matematinio modelio sąvoką — vieną iš svarbiausių OT metodologijoje.) OT projekto fazes sudaro šie veiksmai: aiškiai apibrėžti nagrinėjamosios sistemos tikslus, ištirti darbo aplinką, susipažinti su terminija, žmoniėmis ir daiktais, susijusiais su sprendžiamuoju uždaviniu, išsiaiškinti, kokius metodus gali taikyti vadovas, suformuluoti žodžiu ar kita forma sprendžiamą uždavinį; sudaryti tinkamą loginį, arba matematinį, modelį, kuris susietų sprendžiamojo uždavinio kintamuosius su apribojimais ir efektyvumo matu; rasti sprendinį, optimizuojantį tą efektyvumo matą, t. y. optimalų leistiną sprendinį; matematiniu modeliu gautą sprendinį palyginti su tikrove, siekiant išsiaiškinti, ar iš tikrųjų mes suformulavome ir išsprendėme tą realų uždavinį, kurį pradėjome spręsti; nustatyti, kada keičiasi reali situacija ir kaip dėl to reikia keisti matematinį modelį; pagaliau, svarbiausia — taikymas. Gautąjį sprendinį mes turime pritaikyti praktikoje (ne tik parašyti ataskaitą) ir pažiūrėti, kaip jis tinka realiomis sąlygomis. Kadangi, sudarinėdami tokių uždavinių matematinius modelius, dar nesame pasiekę aukščiausio mokslinio lygio (daugelis netgi mano, kad tai — menas), tai privalome kruopščiai išnagrinėti visus savo sprendinio

---

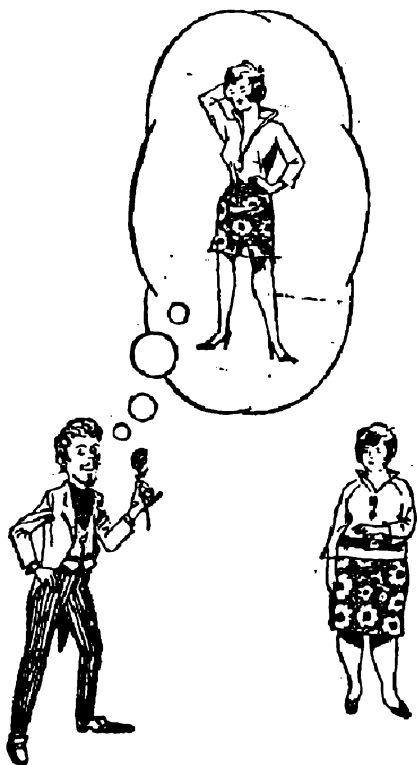
\* R. L. Ackoff. The Development of Operations Research as a Science.— „J. Oper. Res. Soc. Am.“, 1956, vol. 4, No 3.

trūkumus ir tobulinti modelį; tuomet nauji sprendiniai bus realesni ir tikslesni.

Modeliai esti trijų pagrindinių tipų. *Portretinis modelis* yra tiksli originalo kopija. Tai gali būti namo maketas, žvaigždėto dangaus atvaizdas planetariume arba idealiizuotas namų šeimininkės portretas, sukurtas dailininko modeliuotojo. *Analoginis modelis* susieja originalo savybes

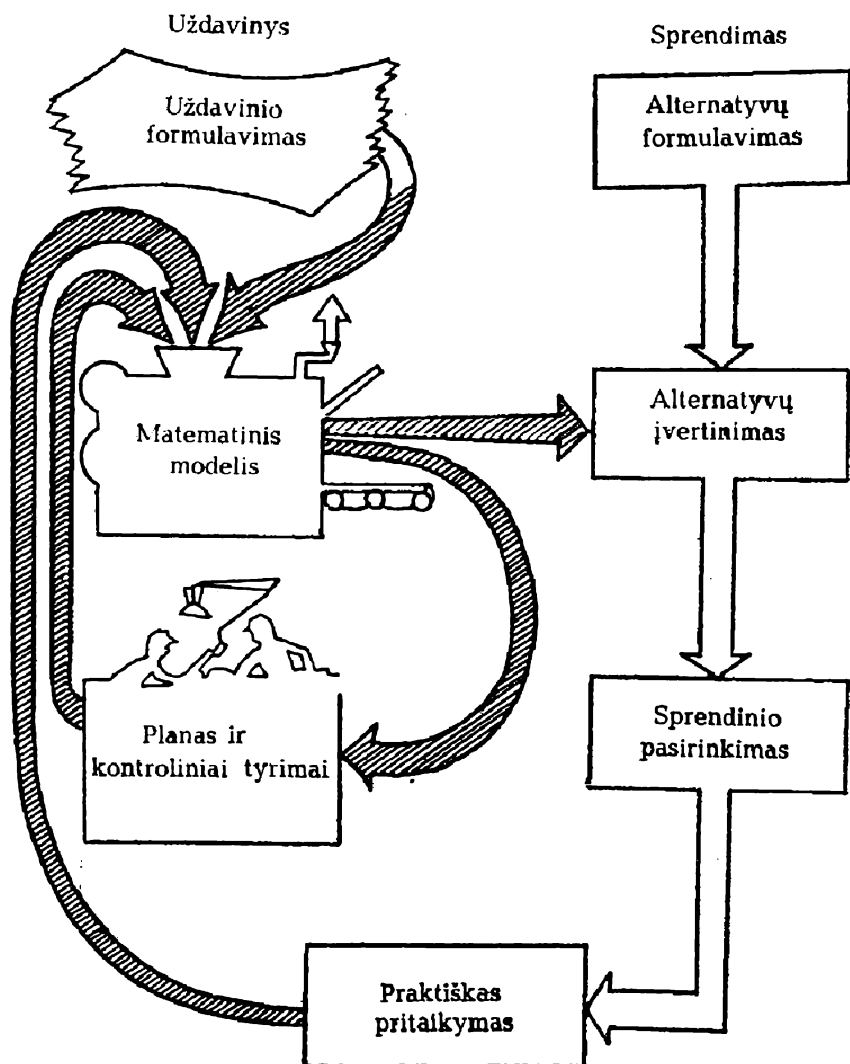
su kitomis vaizdesnėmis ir raiškesnėmis savybėmis. Pavyzdžiui, temperatūra vaizduojama grafiškai, laipsnį laikant ekvivalentišku kuriam nors specialiai parinktam ilgio vienetui. Pagaliau *simbolinis*, arba *loginis matematinis modelis* simboliškai apibūdina tiriamąjį uždavinį arba procesą. Garsioji Einšteino lygtis  $e=mc^2$  simboliškai išreiškia tą faktą, kad energija  $e$ , susijusi su mase  $m$ , yra lygi tos masės ir šviesos greičio kvadrato  $c^2$  sandaugai. Matematinis modelis yra uždavinio vertimas į kiekybinių terminų kalbą. Toliau pamatysime, kad tiesinio programavimo modelis — tai matematinis modelis. Kalbant tiesinio programavimo terminais, matematinis modelis

yra kintamųjų, išteklių, apribojimų ir tikslo funkcijos (efektyvumo mato) sąryšių aibė. OČ metodologijoje matematinis modelis užima svarbiausią vietą. Jis leidžia geriau suprasti sprendžiamąjį uždavinį ir procesus, įvertinti ir palyginti skirtingus sprendinius, padeda įvertinti vieno kintamojo kitimo poveikį visiems kitiems kintamiesiems. Pagaliau matematinis modelis yra, nors ir ne visiškai suprantamas ir todėl šiek tiek paslaptingas, kiekybinis pagrindas intuityviam tiriamojo proceso veikimui patikslinti ir išreikšti skaičiais.



Būtina pabrėžti, kad matematinis modelis yra pirmoji būdinga OT, arba valdymo mokslo, ypatybė. Naudodamiesi matematiniais modeliais, galime taikyti kiek panašią į mokslinę metodologiją tose valdymo srityse, kuriose iki šiol viešpatavo intuicija ir praktinė patirtis. Jų gyvybiškai prireikia ir sandėliuojant įrengimus bei inventorių, ir skirstant išteklius ir masiniame aptarnavime, konfliktinėmis situacijomis, transporte, gamybos procesuose ir kt.

Matematinio modelio vaidmenį valdyme ir OT apibendrintai galima pavaizduoti diagrama, pateikta at-



spausdintame paveiksle\*. Suformulavus uždavinį (tai yra pastrinkus pagrindinį efektyvumo matą), funkcinė matematinio modelio forma jau yra apibrėžta. Kadangi reikia nurodyti, kaip pasirinkti kintamieji yra susiję su pradiniais duomenimis, būtina atlikti kai kuriuos eksperimentus, kurių rezultatai išryškintų teisingą struktūrą. Eksperimentuojant kartais užtenka atskleisti buhalterijos knygą ir iš jos paaimti visą reikalingą informaciją, kitais atvejais tam sunaudojama nemažai jėgų ir lėšų. Ir vienu, ir kitu atveju gautieji rezultatai ir modelio struktūra yra susiję grįžtamuoju ryšiu.

Matematinis modelis kaip tik sieja uždavinį su siūlomu sprendiniu. Dabar pagrindinis mūsų tikslas — naudojantis matematinio modeliu ir efektyvumo matu, įvertinti įvairius uždavinio sprendinius ir iš jų pasirinkti tinkamiausią. Sprendžiant kai kuriuos uždavinius, tai daroma automatiškai — skaičiuojant pagal matematinį modelį. Kaip tik taip daroma tiesiniame programavime. Kitais atvejais tam prireikia nemažai išradingumo ir nuovokos. Kadangi, praktiškai pritaikius gautąjį sprendinį, gali pasikeisti matematinio modelio struktūra, tai privalome grįžti nuo sprendinio prie uždavinio ir visą procesą pakartoti iš naujo. Visa tai tuoj pat pailiustruosime konkrečiu uždaviniu. Sudarysime jo matematinį modelį, kuris šiuo atveju bus tiesinio programavimo modelis.

---

## Transporto uždavinys

### *Didelis dalykas — judėti*

Įmonininkas, besiverčiantis šaldytuvų gamyba ir pardavimu, turi du fabrikus, kurie aprūpina prekėmis tris jo parduotuves. Mėnesio pradžioje iš kiekvienos parduotuvės direktoriaus jis gauna paraišką, kurioje nurodoma, kiek šaldytuvų reikės parduotuvei tą mėnesį. Iš gautų paraiškų įmonininkas apskaičiuoja, kiek iš viso naujų šaldytuvų turės pagaminti abu fabrikai. Paprastumo dėlei tarsime, kad įmonininkas turi pakankamai išteklių (darbo jėgos,

---

\* H. W. Goode. An Application of a Highspeed Computer to the Definition and Solution of the Vehicular Traffic Problem.— „J. Oper. Res. Soc. Am.“, 1957, vol. 5, No 6.

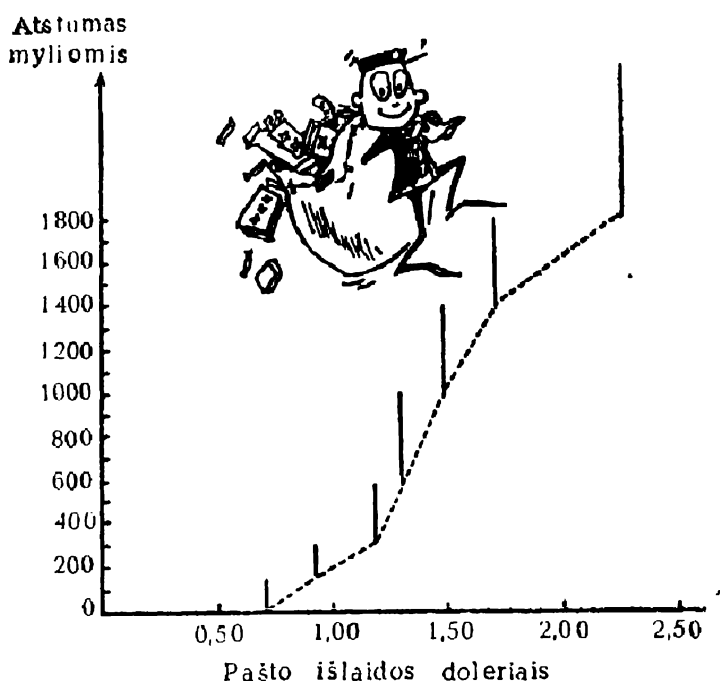
žaliavos ir pan.) visoms paraiškoms patenkinti ir kad tiriamuoju momentu nėra gatavos produkcijos pardavimui. Patį gamybos procesą galima apibūdinti matematiniu modeliu, tačiau dabar mums rūpi visai kas kita. Sakysime, pirmajai parduotuvei, kurią žymėsime  $S_1$ , reikia 10 šaldytuvų, parduotuvei  $S_2$  — 8, o parduotuvei  $S_3$  — 7 šaldytuvų. Taigi iš viso reikia pagaminti 25 šaldytuvus. Sakysime, įmoniminkas nusprendė, kad 11 šaldytuvų pagamins pirmasis fabrikas  $F_1$ , o likusius 14 — antrasis  $F_2$ . Kyla klausimas: po kiek šaldytuvų reikia siųsti iš kiekvieno fabriko į kiekvieną parduotuvę, kad bendroji visų pervežimų išlaidų suma būtų mažiausia? Pagrindinė to uždavinio, vadinamo *transporto uždaviniu*, forma yra viena iš anksčiausių ir dažniausiai pasitaikančių tiesinio programavimo formuluočių.

Mums reikia papildomos informacijos apie transporto apribojimus ir kainas. Sakykime, iš kiekvieno fabriko į bet kurią parduotuvę galima siųsti bet kokių skaičių šaldytuvų; kitaip sakant, kiekvieną fabriką transporto keliai (geležinkeliai, plentai ir pan.) jungia su visomis parduotuvėmis. Toliau tarkime, kad žinome, kiek kainuoja pervežti vieną šaldytuvą iš tų fabrikų į bet kurią parduotuvę. Čia privalome padaryti prielaidą, kad priklausomybės yra tiesinės, nors kai kada tokia prielaida būna abejotina ir kritikuotina. Tiesiškumo, arba proporcingumo, prielaida pagrįsta tuo, kad pervežti vieną šaldytuvą iš  $F_1$  į  $S_1$  kainuoja 10 dolerių, du šaldytuvus — kainuoja jau 20 dolerių, tris — 30 dolerių ir t. t. Tokią prielaidą galima ginčyti, remiantis realiais faktais. Jeigu už 100 dolerių galima pasamdyti sunkvežimį vienam šaldytuvui pervežti, tai, pervežant du šaldytuvus, kiekvieno šaldytuvo transportavimas atsieitų po 50 dolerių (neskaitant pakrovimo ir iškrovimo darbų kainos), o vežant tris šaldytuvus — tik  $33\frac{1}{3}$  dolerio, t. y. pervežimo kaina kinta netiesiškai. Paimkime kitą pavyzdį — siuntinį, įvertintą 3 doleriais ir 80 centų, — ir išsiaiškinkime, kiek kainuos jį pasiųsti iš Atlantik Sičio. Sudarykime lentelę:

| Zona    | Atstumas nuo<br>Atlantik Sičio myliomis | Siuntimo kaina<br>doleriais |
|---------|-----------------------------------------|-----------------------------|
| Vietinė | —                                       | 0,69                        |
| 1       | 50                                      | 0,93                        |
| 2       | 150                                     | 0,93                        |

|   |                   |      |
|---|-------------------|------|
| 3 | 300               | 1,20 |
| 4 | 600               | 1,32 |
| 5 | 1000              | 1,50 |
| 6 | 1400              | 1,71 |
| 7 | 1800              | 1,93 |
| 8 | Daugiau kaip 1800 | 2,25 |

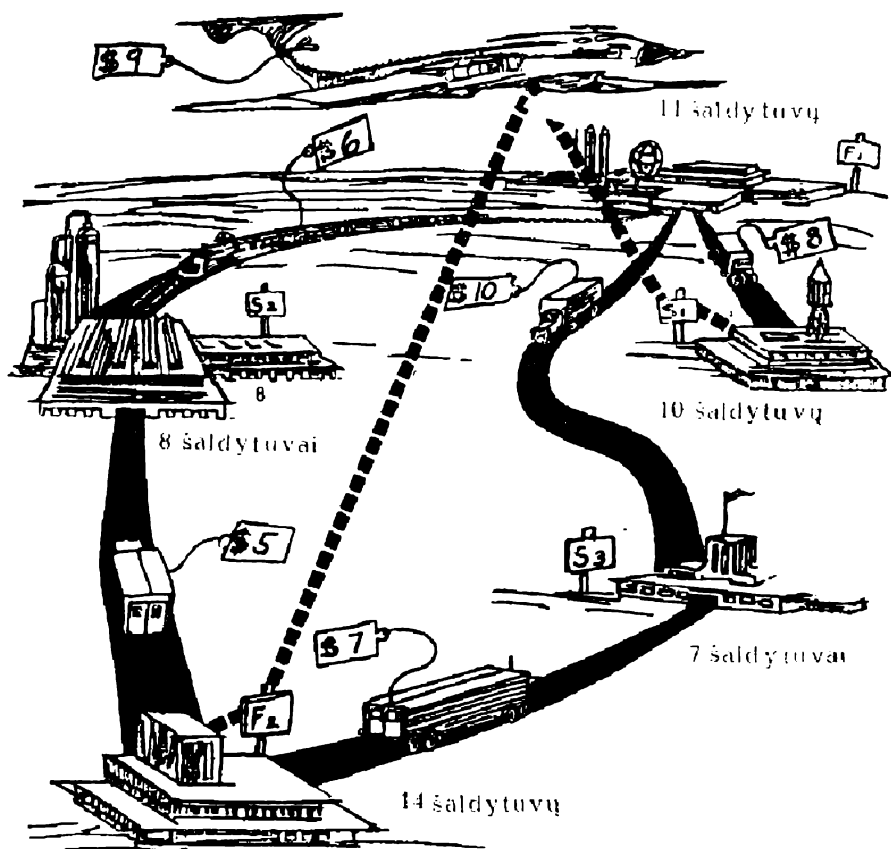
Kainos priklausomybės nuo atstumo grafikas rodo, kad pašto išlaidos kinta netiesiškai. Matome, kad grafikas yra trūki kreivė. Jis sudarytas iš vertikalių, viena su kita



nesujungtų atkarpų. Jeigu kaina, kintant atstumui, tolydžio keistųsi, tai grafikas sutaptų su brūkšnine linija. Dažniausiai įvairiomis transportavimo situacijomis kaina kinta netiesiškai, bet, imant vienokį ar kitokį vidurkį, galima gauti gerą tiesišką artutinybę.

Taigi sprendžiamajame uždavinyje tariame, kad šaldytuvo pervežimo iš bet kurio fabriko į bet kurią parduotuvę kaina yra žinoma ir kad ji kinta tiesiškai. Pateiktame

paveiksle ta kaina parašyta kortelėse. Pavyzdžiui, vieno šaldytuvo pervežimas iš  $F_2$  į  $S_3$  kainuoja 7 dolerius. Paveiksle pavaizduoti keliai, kuriais šaldytuvus iš fabrikų galima pristatyti į parduotuves, ir pateikti visi duome-



nys, reikalingi uždaviniui išspręsti. Tam tikra prasme tai portretinis modelis. Nors tas piešinys ir neišsprendžia uždavinio, vis dėlto jis padeda sudaryti tinkamą matematinį modelį. Apskritai, transporto uždaviniai gali turėti daug leistinų sprendinių. Rasime kai kuriuos iš jų ir kartu pažiūrėsime, kaip tiesinis programavimas pritaikomas šiam uždaviniui.

Ieškosime sprendinio, kuris tenkintų nurodytus apribojimus (11 šaldytuvų reikia pasiųsti iš  $F_1$ , 14 šaldytuvų — iš  $F_2$ ;  $S_1$  turi gauti 10 šaldytuvų,  $S_2$  — 8, o  $S_3$  — 7 šaldy-

tuvus) ir minimizuotų efektyvumo matą — bendrąją visų pervežimų kainą.

Kadangi tą uždavinį norime suformuluoti matematiškai, tai visą informaciją surašysime lentelėje.

|       |       |       |       |    |
|-------|-------|-------|-------|----|
|       | $S_1$ | $S_2$ | $S_3$ |    |
| $F_1$ | \$8   | \$6   | \$10  | 11 |
| $F_2$ | \$9   | \$5   | \$7   | 14 |
|       | 10    | 8     | 7     |    |

Baltieji trikampiai atitinka nežinomą skaičių šaldytuvų, kuriuos reikia pervežti iš atitinkamo fabriko į nurodytą parduotuvę. Norėdami parodyti, kaip lengvai bet kuris patyręs (ir net nepatyręs) gatavos produkcijos tiekimo skyriaus tarnautojas randa sprendinį, pateikiame du galimus variantus:

|       |       |       |       |    |
|-------|-------|-------|-------|----|
|       | $S_1$ | $S_2$ | $S_3$ |    |
| $F_1$ | \$8   | \$6   | \$10  | 11 |
|       | 10    | 1     | 0     |    |
| $F_2$ | \$9   | \$5   | \$7   | 14 |
|       | 0     | 7     | 7     |    |
|       | 10    | 8     | 7     |    |

|       |       |       |       |    |
|-------|-------|-------|-------|----|
|       | $S_1$ | $S_2$ | $S_3$ |    |
| $F_1$ | \$8   | \$6   | \$10  | 11 |
|       | 0     | 7     | 4     |    |
| $F_2$ | \$9   | \$5   | \$7   | 14 |
|       | 10    | 1     | 3     |    |
|       | 10    | 8     | 7     |    |

Skaitytojas be vargo gali rasti ir kitokių sprendinių. Skaičiai, surašyti kairiosios lentelės baltuose trikampeliuose, atitinka tą uždavinio sprendinį, pagal kurį siūloma iš  $F_1$  į parduotuves išsiųsti  $10+1+0=11$  šaldytuvų, o iš  $F_2$  —  $0+7+7=14$  šaldytuvų. Parduotuvė  $S_1$  iš abiejų fabrikų gauna  $10+0=10$  šaldytuvų, parduotuvė  $S_2$  —  $1+7=8$ , o parduotuvė  $S_3$  —  $0+7=7$  šaldytuvus. Antrasis sprendinio variantas yra analogiškas. Tarę, kad pervežimo išlaidos kinta tiesiškai, matome, kad pirmuoju atveju bendroji kaina lygi

$$8 \times 10 + 6 \times 1 + 5 \times 7 + 7 \times 7 = 170 \text{ (dolerių),}$$

o antruoju atveju

$$6 \times 7 + 10 \times 4 + 9 \times 10 + 5 \times 1 + 7 \times 3 = 198 \text{ (doleriai).}$$

Vadinasi, pirmuoju atveju šaldytuvų pervežimo išlaidos yra mažesnės, negu antruoju. Tačiau dar neaišku, ar nėra kito leistino sprendinio, atitinkančio dar mažesnes pervežimo išlaidas. Tarnautojui, nesinaudojančiam tiesinio programavimo metodu, nelieka nieko kita, kaip pasikliauti savo patirtimi ir intuicija. Neperžiūrėdamas visų leistinų sprendinių (jie neperžiūrimi ir taikant tiesinio programavimo metodą), tarnautojas tiesiog pasirenka kurį nors vieną. Suprantama, jis negali tvirtinti, jog rado absoliutų minimumą. Tuo tarpu, remdamiesi tiesinio programavimo metodais, galime teigti, kad tas minimumas egzistuoja ir kad jį galima rasti. Sprendžiant šį uždavinį, išlaidų šaldytuvams pervežti minimumas randamas pirmuoju sprendiniu.

Norint sudaryti transporto uždavinio matematinį modelį, būtina įvesti kai kurias matematines santrumpas. Sakykime,  $x_{11}$  reiškia skaičių (kol kas nežinomą) šaldytuvų, vežamų iš  $F_1$  į  $S_1$ ,  $x_{12}$  — skaičių šaldytuvų, vežamų iš  $F_1$  į  $S_2$ , ir t. t. Apskritai  $x_{ij}$  yra skaičius šaldytuvų, vežamų iš fabriko  $F_i$  į parduotuvę  $S_j$ . Tuos žymėjimus surašysime lentelėje.

Dabar labai paprasta sudaryti matematinį modelį.

Iš  $F_1$  pasiūsta  $x_{11}$ ,  $x_{12}$  ir  $x_{13}$  šaldytuvų, iš viso 11. Panašiai iš  $F_2$  pasiūsta  $x_{21}$ ,  $x_{22}$  ir  $x_{23}$  šaldytuvų, iš viso 14. Kadangi reikalaujama 25 šaldytuvų ( $10 + 8 + 7$ ), o pagaminta jų kaip tik 25 ( $11 + 14$ ), tai iš abiejų fabrikų visi šaldytuvai turi būti išvežti į parduotuves. Vadinasi, bendras skaičius šaldytuvų, pasiūstų iš  $F_1$ , ir reiškiamas lygtimi

$$x_{11} + x_{12} + x_{13} = 11,$$

o iš  $F_2$  — lygtimi

$$x_{21} + x_{22} + x_{23} = 14.$$

Tas sumas gauname, sudėję skaičius, parašytus atitinkamai pirmoje ir antroje lentelės eilutėje.

Kadangi kiekviena parduotuvė gauna tiek šaldytuvų, kiek yra užsakiusi, tai bendras šaldytuvų skaičius, kurį

|       |                 |                 |                  |    |
|-------|-----------------|-----------------|------------------|----|
|       | $S_1$           | $S_2$           | $S_3$            |    |
| $F_1$ | \$8<br>$x_{11}$ | \$6<br>$x_{12}$ | \$10<br>$x_{13}$ | 11 |
| $F_2$ | \$9<br>$x_{21}$ | \$5<br>$x_{22}$ | \$7<br>$x_{23}$  | 14 |
|       | 10              | 8               | 7                |    |

gavo kiekviena parduotuvė (jis randamas, sudedant stulpelių elementus), išreiškiamas šiomis lygtimis:

$$\begin{aligned}x_{11} + x_{21} &= 10 && (\text{parduotuvė } S_1), \\x_{12} + x_{22} &= 8 && (\text{parduotuvė } S_2), \\x_{13} + x_{23} &= 7 && (\text{parduotuvė } S_3).\end{aligned}$$

Kiekvienam skaičių  $x_{ij}$  rinkiniui (čia  $i$  vėl reiškia fabriko numerį, o  $j$  — parduotuvės numerį) visos pervežimo išlaidos lygios

$$8x_{11} + 6x_{12} + 10x_{13} + 9x_{21} + 5x_{22} + 7x_{23} \text{ (dolerių).}$$

Tos lygtys yra pagrindiniai sudaryto matematinio modelio apribojimai. Vienintelis dalykas, kurio dar nepadarėme, — neapribojome  $x_{ij}$  taip, kad tie skaičiai galėtų įgyti tik teigiamas ir nulines reikšmes. Neigiami  $x_{ij}$  reikštų, kad šaldytuvai vežami iš parduotuvės į fabriką, t. y. kad bendras šaldytuvų skaičius yra didesnis už fabrikuose pagamintų šaldytuvų skaičių. Tokios galimybės išvengsime, įvesdami apribojimus  $x_{11} \geq 0$  ( $x_{11}$  didesnis už nulį arba lygus jam),  $x_{12} \geq 0, \dots, x_{23} \geq 0$ , arba apskritai  $x_{ij} \geq 0$ . Tiesiniame programavime tie apribojimai vadinami neneigiamumo sąlygomis. Kadangi norime rasti aibę  $x_{ij}$  reikšmių, tenkinančių lygtis bei neneigiamumo sąlygas ir minimizuojančių bendrąją kainą, tai turime šitokį matematinį modelį — aptariamojo transporto uždavinio tiesinio programavimo modelį:

reikia rasti aibę neneigiamų skaičių  $x_{ij}$ , kurie minimizuoja reiškinį

$$8x_{11} + 6x_{12} + 10x_{13} + 9x_{21} + 5x_{22} + 7x_{23} \text{ (dolerių)}$$

ir tenkina apribojimus

$$\begin{aligned}x_{11} + x_{12} + x_{13} &= 11, \\x_{21} + x_{22} + x_{23} &= 14, \\x_{11} + x_{21} &= 10, \\x_{12} + x_{22} &= 8, \\x_{13} + x_{23} &= 7.\end{aligned}$$

Pirmasis iš anksčiau pateiktųjų sprendinių, t. y.  $x_{11} = 10$ ,  $x_{12} = 1$ ,  $x_{13} = 0$ ,  $x_{21} = 0$ ,  $x_{22} = 7$  ir  $x_{23} = 7$  tenkina tas lygtis;

kaip jau sakėme, tą sprendinį atitinka tikslo funkcijos minimumas, lygus 170 dolerių.

Kiekvieną fabriką ir kiekvieną parduotuvę atitinka lygtis tų kintamųjų, kurie susiję su tuo fabriku ar parduotuve. Tos lygtys, taip pat ir tikslo funkcija yra tiesinės, nes jos yra tiesiog kintamųjų sumos. Bendras kintamųjų skaičius yra lygus fabrikų ir parduotuvių skaičių sandaugai, tiriamuoju atveju —  $2 \times 3 = 6$ . Lygčių skaičius yra lygus fabrikų ir parduotuvių skaičių sumai; nagrinėjamuoju atveju jis lygus 5. Transporto uždaviniai gali būti labai griozdiški, tačiau dauguma elektroninių skaičiavimo mašinų pajėgia susidoroti su skaičiavimo procedūromis (algoritmais), neišvengiamomis, sprendžiant didelės apimties uždavinius.

Anksčiau parašytoji lygčių sistema matematiniu požiūriu yra gana įdomi. Pirmiausia, joje yra viena nereikalinga lygtis, išplaukianti iš kitų lygčių. Pavyzdžiui, jeigu išbrauktume pirmąją lygtį, tai ją galėtume vėl sudaryti, antrąją lygtį atimdami iš trijų paskutinių lygčių sumos. Tačiau svarbiausia yra tai, kad, sprendžiant tą uždavinį standartiniais tiesinio programavimo metodais, kintamųjų  $x_{ij}$  reikšmės optimaliame sprendinyje yra sveikieji skaičiai. Iki šiol tylomis buvo tariama, kad skaičiai  $x_{ij}$  yra sveikieji (juk neįmanoma pervežti  $3\frac{3}{4}$  šaldytuvo!). Galima griežtai įrodyti, kad tuo atveju, kai fabrikų ir parduotuvių skaičius reiškiamas sveikaisiais skaičiais, transporto uždavinio optimalus sprendinys taip pat bus išreikštas sveikaisiais skaičiais. Galimybė transporto uždavinį išspręsti sveikaisiais skaičiais išplaukia iš savotiškos tą uždavinį atitinkančio matematinio modelio lygčių struktūros.

Tiriamąjį modelį aprašėme, remdamiesi konkrečiu pavyzdžiu, todėl buvome priversti kalbėti apie fabrikus, parduotuves ir šaldytuvus, tačiau tos detalės neturi reikšmės. Visi samprotavimai lieka teisingi ir bendresniu atveju, kai nagrinėjimo sfera apima siuntimo punktus (fabrikus), paskirties punktus (parduotuves) ir pervežtinius vienerūšius vienetus (šaldytuvus), taip pat tam tikrą matą, kurį reikia minimizuoti (bendrąją pervežimo kainą). Šią mintį tuoj pat pailiustruosime, pateikdami kitą tiesinės programos, vadinamos transporto modeliu, pavyzdį.

Naujajame transporto uždavinyje kalbama apie tam tikro krovinio pervežimą iš sandėlių į žinomus paskirties punktus. Krovinio kiekis kiekviename sandėlyje yra ribotas. Reikalingą krovinio kiekį į paskirties punktus galima pristatyti įvairiais būdais. Uždavinys — rasti būdą ne tik visiems poreikiams patenkinti, bet ir minimizuoti tam tikrą matą, pavyzdžiui, kainą. Galima minimizuoti vieną iš šių tikslo funkcijų: bendrą kainą doleriais, bendrą ilgį visų kelių, kuriais pristatomi kroviniai, arba bendrą visų tranzitinių reisų skaičių.

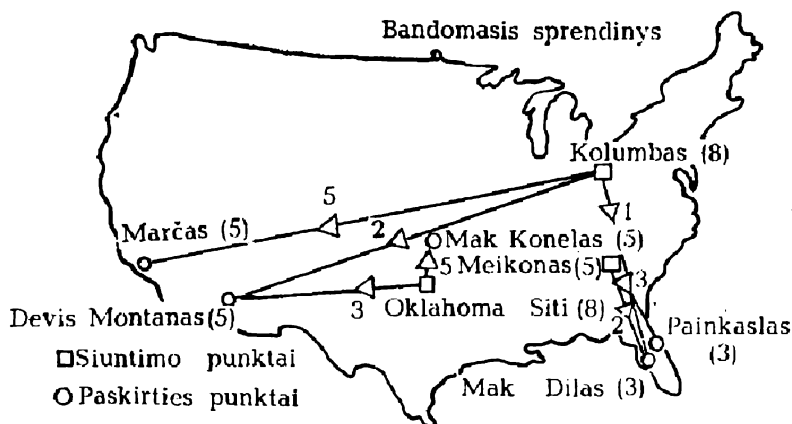
Sakykime, Lekburno aviacijos bazėje, esančioje Kolumbe, buvo išbandyta didelė lėktuvų įrangos partija. Vienas tos įrangos komplektas sveria vieną toną. Tą įrangą reikia išbandyti kitose bazėse. Marčo, Devis Montano ir Mak Konelo bazėms reikia išsiųsti po penkis tos įrangos komplektus, o Painkaslo ir Mak Dilo bazėms — po tris. Iš reikalingų 21 komplekto Kolumbe yra 8, Oklahomos sandėlyje — 8 ir Meikone — 5. Įranga į paskirties punktus pristatoma lėktuvais. Visi reikalingi duomenys, įskaitant atstumus tarp siuntimo ir paskirties punktų, surašyti šioje lentelėje:

| Siuntimo punktai | Turimų komplektų skaičius | Paskirties punktams           |        |                |             |            |
|------------------|---------------------------|-------------------------------|--------|----------------|-------------|------------|
|                  |                           | Mak Dilas                     | Marčas | Devis Montanas | Mak Konelas | Painkaslas |
|                  |                           | reikalingų komplektų skaičius |        |                |             |            |
|                  |                           | 3                             | 5      | 5              | 5           | 3          |
|                  |                           | Atstumas myliomis             |        |                |             |            |
| Oklahoma Sitis   | 8                         | 938                           | 1030   | 824            | 136         | 995        |
| Meikonas         | 5                         | 346                           | 1818   | 1416           | 806         | 296        |
| Kolumbas         | 8                         | 905                           | 1795   | 1590           | 716         | 854        |

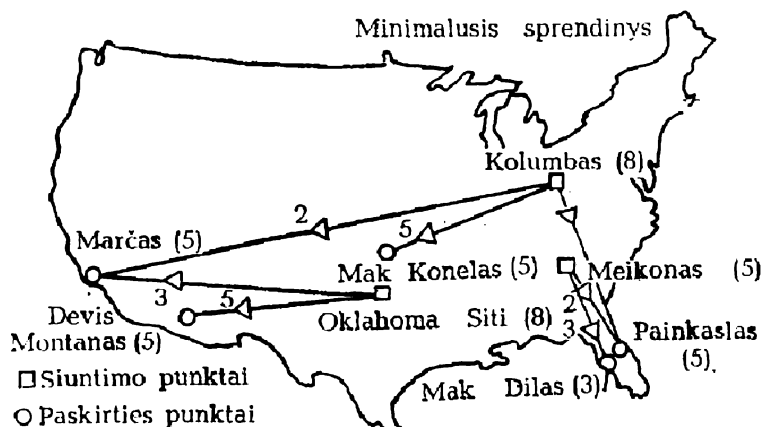
Reikalaujama minimizuoti tonmylių skaičių. Optimaliausias tuo atžvilgiu sprendinys pateiktas 27 puslapyje. Skaitytojui naudinga pačiam pamėginti rasti geriausią pervežimo būdą.

Galima ieškoti, pavyzdžiui, šitaip. Stengsimės patenkinti bazių poreikius iš artimiausių šaltinių. Sakykime, iš sandėlio, esančio Oklahoma Sityje, Mak Konelo bazė

gauna visus 5 komplektus, o Devis Montano bazė — 3 iš 5 reikalaujamų. Painkaslo ir Mak Dilo bazių, esančių Floridoje, poreikius, kiek įmanoma, reikia patenkinti iš Meikono bazės. Sakykime, Painkaslas gauna visus 3 komplektus,



o Mak Dilas — 2 iš 3 reikalaujamų. Likusius įrangos komplektus reikia pristatyti iš Kolumbo. Šioje bandomojoje schemoje bendrasis tonmylių skaičius lygus 17 792 (žr. 26 p.).



Yra daug kitų leistinų sprendinių. Minimalusis sprendinys (žr. paveikslą), rastas tiesinio programavimo metodais, yra 16 864 tonmylės. Spręsdami tiesinio programavimo metodais, pradedame nuo bet kurio sprendinio (pavyzdžiui, nuo ką tik gauto) ir randame vis geresnius ir geresnius sprendinius, kol galų gale randame optima-

liausia. Be to, tie metodai laiduoja, kad minimumą tikrai rasime.

Jeigu skaitytojas mėgintų patenkinti dalį Mak Konelo bazės poreikių, pristatydamas įrangą iš Oklahoma Sičio (t. y. naudotąsi, kaip buvo daroma, sudarant pirmą bandomąjį sprendinį, artimiausiu sandėliu, kuriame yra reikalinga įranga), tai bendrasis tonmylių skaičius būtų didesnis. Tai dar kartą patvirtina, kad sprendimas, kuris iš pirmo žvilgsnio atrodo natūraliausias, ne visada yra pats geriausias.

Transporto uždavinio ir įvairiausių jo variantų nagrinėjimas galėtų būti ištiso traktato tema; tai svarbus valdymo mokslo skyrius. Toliau mums dar ne kartą pasitaisy tokių uždavinių. O dabar tik pasakysime, kad matematiniai transporto uždavinio modeliai gali būti labai įvairūs: reikia turėti omenyje jaunosios matematikų kartos išradingumą ir turtingą vaizduotę. Tačiau man norėtųsi atkreipti skaitytojo dėmesį į tai, kad net mėgindami paprastai, logiškai formuluoti uždavinį, galime gerokai priartėti prie teisingo matematinio modelio. Tiesinio programavimo sferoje ši mintis yra daug kartų ryškiai patvirtinusi. Čia sudėtingus ir ypač painius klausimus paprastai pavyksta išspręsti be juodosios magijos paslaugų.

Keliaujant po Tiesinio Programavimo Šalį, labai svarbu retkarčiais stabtelėti ir iš perspektyvos pažvelgti į nuėitą kelią, pamąstyti apie jo sąryšį su gretimomis sritimis. Skyriaus pabaiga — tam tinkamiausia vieta, todėl, mano nuomone, čia būtina pareikšti štai ką.

Mano (ir ne vien mano) žavėjimasis tiesiniu programavimu šiek tiek atvėsta, kai prisimenu, kokius sunkumus reikia įveikti, formuluojant daugelį operacinių uždavinių. Sunkumai atsiranda ir renkantis reikalingus apribojimus, ir derinant prieštaringas tendencijas, ir svarstant, kokių duomenų reikia, ir tų duomenų ieškant. Negalima pamiršti ir psichologinių bei politinių faktorių, būdingų kiekvienai situacijai, jei tik gali kilti įtarimas, kad matematika ir matematikai ketiną pakeisti tuos, kurie „iš tikrųjų“ išmano tą uždavinį.

Nors tiesinis programavimas ir jo apibendrinimai sudaro svarbią taikomosios matematikos šaką, bet jie yra tik platesnės valdymo mokslo srities — operacijų tyrimo — sudėtinė dalis. Anksčiau buvo manoma, kad pagrindinė

OT paskirtis — optimizuoti dideles jungtines sistemas, panašias į gamybos ir skirstymo sistemą. Tačiau iš tikrųjų šiuo būdu galima spręsti uždavinius, panašius į supaprastintą transporto uždavinį ir aprašančius tik posistemų posistemius. Taip yra dėl įvairių (kol kas dar nepašalintų) priežasčių: nesugebame sudaryti tikslaus matematinio modelio, atspindinčio kompleksinę tarpusavio sąveiką, ir neturime skaičiavimo technikos, kuri padėtų spręsti tokius uždavinius. Vadinasi, supaprastintas tiesinio programavimo uždavinių aptarimas, pateikiamas šioje knygelėje, turi tuos pačius trūkumus, būdingus posistemiams, kaip ir visas OT. Mes jau įveikiame svarbius uždavinių poklasius, bet didelių sistemų optimizavimo uždaviniai vis dar laukia sprendėjų.

Tad kokios perspektyvos? Numanoma, kad visiškai optimizuoti kompleksines sistemas arba nors sėkmingai atlikinėti operacijas bus galima tik išvysčius informacines sistemas, kurios, pasitelkusios ESM, atspindėtų tiramosios operacinės sistemos momentinę būklę. Tokioje informacinėje sistemoje atitinkami matematiniai modeliai ir skaičiavimo metodai (panašūs į tiesinį programavimą) turi būti susieti su euristiciais metodais, besiremiančiais ESM ir įgalinančiais žmogų, pasitelkusį ESM, išnagrinėti tuos uždavinio klausimus, kurių nepavyksta aprašyti tikslia matematikos kalba. Pavyzdžiui, sprendami gamybos ir skirstymo uždavinį (panašų į tą, kurį sprendė įmonininkas, besiverčiantis šaldytuvų gamyba ir realizavimu), mes nepajėgtume, gal būt, sudaryti matematinio modelio, parodančio, kaip reikia skirstyti resursus bei įrangą ir kokia eile gaminti produkcijos vienetus. Tačiau, turėdami gatavą produkciją, sugebame pritaikyti transporto modelį ir į paskirties punktus produkciją galime pasiųsti optimaliu būdu. Šiuo atveju informacinė sistema parodytų gamybos būklę ir produkcijos, kurią būtina pagaminti, kiekį. Naudodamasis euristiciais metodais, įmonininkas galėtų sudaryti keletą gamybos modelio variantų, atitinkančių tam tikras sąlygas ir prielaidas; apskaičiuoti kiekvieno varianto kainą; naudodamasis transporto modeliu, pagal kiekvieną variantą apskaičiuoti minimalią pervežimo kainą; pagaliau, pasirinkti tą variantą, pagal kurį bendra išlaidų suma yra mažiausia. Šitaip darydami, tikrojo išlaidų minimumo nesurasime, nes tarėme, kad visą gamybos procesą optimizuojančio modelio neturime. Tiesa, kai ESM skai-

čiuoja labai greitai, įmonininkas, prieš pradėdamas realią gamybą, aprašytuoju metodu gali išmėginti daug įvairių gamybos variantų (įskaitant ir tą, kuris dažniausiai taikomas jo fabrikuose). Vadinasi, įmonininkas galės optimizuoti visą procesą bent tiek, kiek įmanoma aprašytuoju metodu.

Matome, kad tokia hipotetiniame jungtiniame informacijos, optimizacijos ir euristikos metode optimizavimo elementas tėra vienas būdingas bruožas bendrųjų kontrolės metodų, taikomų kompleksiniams uždaviniams spręsti. Matyt, ateityje tiesinis programavimas bus panašių metodų pagrindas, kaip jis jau yra pagrindinis optimizacijos metodas, sprendžiant daugelį smulkesnių realių uždavinių.

---

## 2. Uždavinių formulavimas

(Tęsiame kelionę ir sužinojome, kaip reikia sudaryti meniu, aplankome fermą ir netikėtai sutinkame keistas būtybes, gyvenančias Tiesinio Programavimo Šalyje.)

Pagrindinis tikslas, kurio siekiame, sudarinėdami kokio nors operacinio uždavinio tiesinio programavimo modelį,— iš pradinių duomenų numatyti optimalų to uždavinio sprendinį. Tokiu atveju tariama, kad realioje situacijoje (t. y. tame praktiniame uždavinyje, kurį norime išspręsti) įmanoma teisingai pasirinkti kintamuosius bei apribojimus ir tinkamai atlikti veiksmus su jais. Daugečiu atvejų (praktiškai beveik visada) mūsų sugebėjimai tiksliai aprašyti *tikrą, praktinį uždavinį* būna gana abejotini. Tačiau paaiškėja, kad čia, kaip ir kitose žmogaus veiklos srityse, kompromisai, nuovoka ir netgi tam tikra laisvė, sąlygojama mūsų žinių netikslumo, padeda geriau suprasti tiriamąjį procesą ir, kaip mes tikimės, sudaryti modelį, pagal kurį galima rasti geresnį sprendinį, tinkamą taikyti praktikoje. Tačiau modelis, nors ir pretenduojantis tiksliai aprašyti kokio nors fabriko gamybinės galimybes, iš tikrųjų tai gali padaryti tik tam tikrose ribose. Mūsų hipotezės apie būsimąją produkciją, kuriomis remdamiesi planuojame pervežimą, pardavimą ir kitus poreikius, pagrįstos numatomais arba išmatuotais gamybos tempais, esama darbo jėga, resursais ir t. t. Tačiau viso-

kie atsitiktinumai, kurių pasitaiko tikrovėje (tai — sugaištis, tai — trūkumas), turi įtakos tikrajam produkcijos kiekiui. Jeigu modelis matematiškai „protingai“ aprašo realųjį uždavinį, tai planai, pagrįsti numatoma produkcija, nenuves mūsų pernelyg toli į šalį ir praktiškai padės mums veikti efektyviau, vadinasi, ir pelningiau.

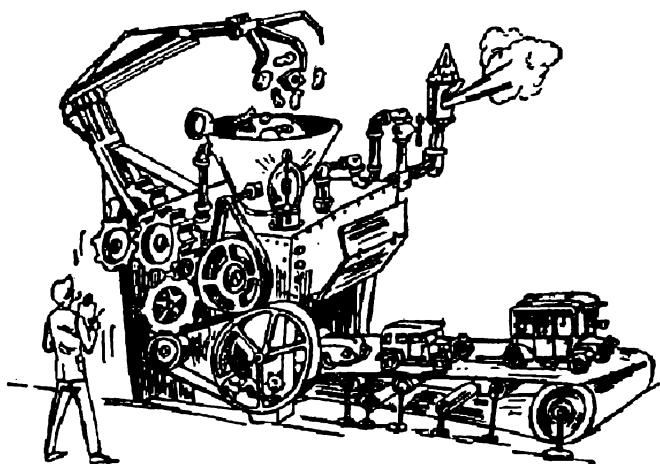
Formuluodami uždavinius tiesinio programavimo terminais, turime apsidrausti nuo kaltinimų, kad mes esą įrankį laikome svarbesniu už patį darbą ir, kai įrankis netinkas, keičiame patį darbą, siekdami jį pritaikyti turimam įrankiui. Tos galimos pinklės vis dėlto netrukdo ir neturi mums trukdyti, kai reikia suprastinti modelį, teisingai atspindintį pačią dalyko esmę. Mūsų matematinių modelių atsakymai turi būti aiškūs tiems, kas atsako už tiriamuosius procesus. Juk tiems žmonėms, kuriuos ne visiškai teisingai vadina „valdytojais“, gautuosius rezultatus reikės panaudoti praktikoje. Tie rezultatai turi pagerinti (tinkamai parinkto efektyvumo mato atžvilgiu) mūsų veiksmus.

Kaip ieškome tiesinio programavimo uždavinio formulotės? Kokiose vietose slypi pavojus? Kaip randame reikalingus rezultatus? Į tuos ir panašius klausimus neįmanoma atsakyti griežtai ir aiškiai. Daugiausia, ką galiu padaryti,— pakalbėti apie tai, kad skaitytojui paaiškėtų jų prasmė.

Jau sakėme, kad transporto uždaviniu iš pradžių buvo pavadintas uždavinys apie prekių pervežimą iš fabrikų į parduotuves, o bendrasis transporto uždavinys yra apie bet kokius siuntimo ir paskirties punktus. Tą aplinkybę būtina ypatingai pabrėžti. Keliais pavyzdžiais parodysime, kaip tas uždavinys atrodo įvairiose situacijose. Skaitytojui (kuriam jau metas suprasti, kokią sunkią naštą ant jo pečių užkrovė autorius) nederėtų būti ribotam ir galvoti, kad galutinį matematinį modelį galima pritaikyti tik tai konkrečiai situacijai, kuriai jis sudarytas. Istorija, kurią papasakosime, paaiškins iškeltą mintį.

Tik atsiradus tiesiniam programavimui, maždaug apie 1953 metus, viename straipsnyje uždavinio formulavimo esmė buvo aiškinama pavyzdžiu apie optimalų produktų mišinio sudarymą. Fabriko, perdirbančio riešutus, savininkas nori sumaišyti trijų rūšių riešutus. Kiekviena porcija sudaroma iš kešių, funduko ir kedro riešutų, laikantis tam tikrų sąlygų. Pavyzdžiui, vienoje porcijoje turi būti

ne mažiau kaip 50% kešiu ir ne daugiau kaip 25% kedro riešutų. Atsižvelgdamas į fabriko galimybes, įmonininkas turimus riešutų išteklius nori taip panaudoti, kad pelnas būtų maksimalus. Straipsnyje buvo smulkiai paaiškintas pats uždavinys, pateiktas jo matematinis modelis ir skai-



tinis pavyzdys. Tuo pavyzdžiu ir straipsniu susidomėjo labai daug skaitytojų. Ir štai vieną pirmųjų konsultantų valdymo klausimais pasikviečia stambios Detroito automobilių kompanijos direktorius, norėdamas sužinoti, kur galima susipažinti su naujuoju metodu, vadinamu tiesiniu programavimu. Konsultantas jam rekomendavo kaip tik tą straipsnį, kuriame pateiktas uždavinys apie riešutus. Tačiau, užėjęs po kelių savaičių pasiteirauti, ar nereikia jo paslaugų, konsultantas susilaukė gausių priekaištų, nes direktorius tik veltui sugaišęs laiką. Pasirodo, direktorius, perskaitęs ir išstudijavęs viską, kas buvo susiję su uždaviniu apie riešutus, pasijuto esąs geras rytietiško saldumynų specialistas, bet jis juk turėjo gaminti... automobilius. Direktorius nemokėjo „išversti“ sąvokų, susijusių su uždaviniu apie riešutus, į savo gamybos kalbą. Jis nesugebėjo „susieti“ vieno su kitu. Aš įsitikinęs, kad šiuolaikinis skaitytojas yra įžvalgesnis. Mažų mažiau- siai jį jau įspėjau.

Norėdamas parodyti, koks lankstus yra tiesinio programavimo modelis ir kaip lengva jį pritaikyti įvairiomis situacijomis, toliau pateiksiu keletą uždavinių, dabar jau tapusių klasikiniiais ir aprašytų tiesinio programavimo lite-

ratūroje. Matematinio požiūriu visi tie uždaviniai yra analogiški, visi jie išsprendžiami standartiniu tiesinio programavimo skaičiavimo metodu — *simplekso metodu*. Tam metodui pagrįsti reikėtų gana sudėtingų matematinių sąvokų, bet šioje knygoje jų nenagrinėsime\*. Mano tikslas — supažindinti skaitytoją su esmingiausiais tiesinio programavimo bruožais, o ne padaryti jį tos srities specialistu. Vadinasi, kalbėdami šiuo klausimu, apie skaičiavimą beveik neužsiminsime, palikdami tą temą kitam skyriui ir priedui.

---

## Dietos uždavinys

*Norint sužinoti, ar skanus pudingas,  
reikia jį suvalgyti*

Kiekvienas, kas nori gyventi, neišeidamas iš savo biudžeto rėmų, lėšas gali skirstyti labai įvairiai. Taupi šeiminkė tam tikrą sumą atideda mokesčiui už butą, o kitą dalį skiria rūbams, maistui, pramogoms, transportui ir t. t. Lengva apskaičiuoti, kiek lėšų reikia skirti reikalams, panašiams į mokesį už butą, nes jos yra pastovios. O štai, skirstydami pinigus maistui ir pramogoms, dažniausiai remiamės ankstesne patirtimi ir atsitiktinėmis aplinkybėmis. Konkretus lėšų paskirstymas vis dėlto priklauso ir nuo efektyvumo matų, kurį pasirenka šeiminkė, atsižvelgdama į visas išlaidas. Sprendžiant tokius uždavinius, gana sunku ne tik optimizuoti, bet ir nustatyti bendrą efektyvumo matą. Todėl dažniausiai stengiamasi tą uždavinį suskirstyti į smulkesnius ir lengviau išsprendžiamus optimumo uždavinius; kiekvieną iš jų atitinka tam tikras efektyvumo matas ir su juo susiję apribojimai. Išnagrinėsime, pavyzdžiui, vieną nesudėtingą uždavinį, kurį sprendžia šeiminkė, ketindama pamaitinti savo šeimą. Šį uždavinį labai supaprastinsime (panašiais atvejais taip darysime ir toliau), būtent, tirsime tik tą atvejį, kai šeiminkė gamina pusryčius savo vaikams. Pamiršusi biudžetą, rūpestinga šeiminkė stengiasi, kad pusryčiai būtų kuo maistingesni. Žvilgterėjusi į žinyną, kuriame surašyti duo-

---

\* Skaitytojui, pageidaujančiam susipažinti su matematine to klausimo puse, rekomenduojame perskaityti šios knygos priedą ir jos autoriaus vadovėlį: С. Гасс. Линейное программирование (методы и приложения). М., Физматгиз, 1961.

menys apie vitaminus ir kalorijas, ji sužino, kad vaikai per pusryčius turi gauti ne mažiau kaip 1 mg tiamino, 5 mg niacino ir 400 kalorijų. Pusryčiams ji turi dviejų rūšių kruopų: *K* ir *C*. Galima pasirinkti arba vieną rūšį, arba kitą, arba pagaliau abiejų rūšių mišinį. Gražiose



etiketėse ant visų dėžučių, be kita ko, parašyta, kad 1 uncijoje *K* yra 0,10 mg tiamino, 1 mg niacino ir 110 kalorijų, tuo tarpu 1 uncijoje *C* yra 0,25 mg tiamino, 0,25 mg niacino ir 120 kalorijų. Matome, kad labai lengva rasti to uždavinio sprendinius, t. y. sudaryti įvairius meniu. Reikalingą maistingųjų medžiagų kiekį gausime, suvalgę arba 10 uncijų *K*, arba 20 uncijų *C*. O kiek tokie pusryčiai kainuoja, ar neviršytas biudžetas? Šeimnininkė turi taip sudaryti meniu, kad vaikai gautų būtiną maistingųjų medžiagų kiekį, o išlaidos būtų mažiausios. Jeigu viena uncija *K* kainuoja 3,8 cento, o uncija *C* — 4,2 cento, tai dviejų pasiūlytųjų sprendinių kaina atitinkamai lygi 38 ir 84 centams. Pagal pirmąjį sprendinį (suvalgyti 10 uncijų *K*) vaikai gaus 1 mg tiamino, 10 mg niacino ir 1100 kalorijų, o pagal antrąjį (suvalgyti 20 uncijų *C*) — 5 mg tiamino, 5 mg niacino ir 2400 kalorijų. Pagal pirmąjį sprendinį gaunamas reikiamas kiekis tiamino, bet per daug kitų sudedamųjų dalių; pagal antrąjį — tiek, kiek reikia

niacino, bet per daug visų kitų. Jeigu vaikai pusryčiams prašys tik *K*, tai šeimnininkė negalės imti *K* mažiau kaip 10 uncijų, nes mažesniame *K* kiekyje nebūtų nė 1 mg ti-amino. Jeigu vaikai norės tik *C*, tai jie turės jų suvalgyti 20 uncijų, nes priešingu atveju negautų pakankamai nia-cino. Vienintelis leistinas variantas, kuris skiriasi nuo ankstesniųjų, yra abiejų rūšių kruopų mišinys (šeiminin-kė turi apskaičiuoti, ar yra mišinys, turįs pakankamai maistingųjų medžiagų ir pigesnis kaip 10 uncijų *K*). Iš tikrųjų ji norėtų sudaryti patį pigiausią mišinį, tenkinantį „maistingumo apribojimus“. Tačiau- kaip tai padaryti? Ir štai čia į pagalbą ateina tiesinis programavimas!

Anksčiau pateiktuose samprotavimuose, kalbėdami apie dviejų rūšių produktų mišinį, tylomis tarėme, kad kai ku-rios priklausomybės yra tiesinės. Pirmiausia, laikėme, kad gautų maistingųjų medžiagų kiekis yra proporcingas su-valgyto maisto kiekiui. Pavyzdžiui, vienoje uncijoje *K* yra 110 kalorijų, o 2 uncijose — 220 kalorijų ir t. t. Panašiai maistingųjų medžiagų, esančių vienos rūšies maiste, kie-kis susideda su analogišku kiekiu, esančiu kitos rūšies maiste. Pavyzdžiui, 1 uncijoje *K* ir 1 uncijoje *C* iš viso yra  $110 + 120 = 230$  kalorijų. Uždavinio apribojimai yra su-siję su minimumu maistingųjų medžiagų, kurias turi gauti vaikai; jeigu pagal kurį nors sprendinį jų gauname dau-giau, negu būtina, tai toks sprendinys irgi tinkamas. Pa-vyzdžiui, 10 uncijų *K* turi 1100 kalorijų, nors jų reikia tik 400. Norėdami sudaryti tiesinio programavimo modelį, vi-sus tiriamojo uždavinio duomenis surašysime šitokioje lentelėje:

|                 | Kiekis 1 uncijoje<br><i>K</i> | Kiekis 1 uncijoje<br><i>C</i> | Poreikis |
|-----------------|-------------------------------|-------------------------------|----------|
| Tiaminas        | 0,10 mg                       | 0,25 mg                       | 1        |
| Niacinas        | 1,00 mg                       | 0,25 mg                       | 5        |
| Kalorijos       | 110,00                        | 120,00                        | 400      |
| 1 uncijos kaina | 3,8 cento                     | 4,2 cento                     |          |

Rašydami uždavinio apribojimus, atkreipsime dėmesį, kad bendras kiekvieno vitamino ir kalorijų kiekis kairiuo-siuose stulpeliuose turi būti ne mažesnis ( $\geq$ ) už dydį,

esantį dešiniajame stulpelyje. Vadinas, kiekvieną sudedamąją dalį atitinka vienas apribojimas. Pavyzdžiui,  $K$  uncijų pirmosios rūšies kruopų turi  $0,1K$  mg tiamino, o  $C$  uncijų antrosios rūšies kruopų turi  $0,25C$  mg šios medžiagos. Todėl uždavinio sprendinys turi tenkinti nelygybę

$$0,1K + 0,25C \geq 1.$$

Panašiai atsižvelgdami į niacino kiekį, gauname nelygybę

$$1,00K + 0,25C \geq 5,$$

o atsižvelgdami į kalorias, — nelygybę

$$110,00K + 120,00C \geq 400.$$

Vienos ir kitos rūšies maisto kiekis turi būti išreiškiamas arba teigiamu skaičiumi, arba nuliu, todėl  $K \geq 0$  ir  $C \geq 0$ . Pagaliau bendroji meniu kaina išreiškiama formule

$$3,8K + 4,2C.$$

Surinkę visas tas priklausomybes į vieną vietą, matome, kad mūsų šeimnininkės uždavinį galime suformuluoti šitaip: reikia rasti dydžius  $K$  ir  $C$ , minimizuojančius bendrąją kainą

$$3,8K + 4,2C$$

ir tenkinančius šias nelygybes:

$$0,10K + 0,25C \geq 1,$$

$$1,00K + 0,25C \geq 5,$$

$$110,00K + 120,00C \geq 400,$$

$$K \geq 0,$$

$$C \geq 0.$$

Skaitytojas gali lengvai patikrinti, kad du anksčiau sudaryti meniu  $K=10$ ,  $C=0$  ir  $K=0$ ,  $C=20$  tenkina parašytąsias nelygybes. Minimalusis sprendinys gaunamas, imant  $K=4\frac{4}{9}$  uncijos ir  $C=2\frac{2}{9}$  uncijos; tada bendroji kaina lygi  $26\frac{2}{9}$  cento. Šiuo atveju vaikai gauna 1 mg tiami-

no, 5 mg niacino (tiksliai tiek, kiek reikia) ir  $755 \frac{5}{9}$  kalorijas (perteklius —  $355 \frac{5}{9}$  kalorijų\*).

Tą paprastą dietos, arba meniu sudarymo, uždavinį galima apibendrinti. Naujajame uždavinyje reikalaujama sudaryti visos paros dietą, tenkinančią maistingųjų medžiagų poreikius ir minimizuojančią bendrą kainą. Pirmą kartą toks uždavinys buvo suformuluotas keturiasdešimtųjų metų pradžioje, dar nežinant tiesinio programavimo metodų. Ekonomistas D. Stigleris suformulavo tą uždavinį, imdamas 77 produktų rūšis ir 1939 metų kainas. Uždaviniui spręsti Stigleris pritaikė bandymų ir klaidų metodą. Kruopšti analizė ir gerai išvystyta intuicija padėjo jam rasti tokias produktų rūšis ir kiekius, kad būtų reikiamas maistingųjų medžiagų minimumas vienai dienai, o tokio meniu kaina būtų labai žema, nors ir ne minimali. Jo sprendinyje buvo tik penki produktai, kurių bendra metinė kaina — 39,93 dolerio. Tai — kvietiniai miltai, sausas pienas, kopūstai, špinatai ir sausi žirniai. Minimalios kainos dieta, rasta tiesinio programavimo metodais, susideda iš devynių produktų: kvietinių miltų, kukurūzų miltų, sauso pieno, arachio aliejaus, lydytų kiaulienos taukų, jautienos kepenų, kopūstų, bulvių ir špinatų; ją atitinka truputį žemesnė metinė kaina, lygi 39,67 dolerio. Tokios dietos yra nepaprastai pigios, tačiau jos visiškai nesuvirškinamos, todėl jomis gali žavėtis nebent kokios nors koncentracijos stovyklos vyriausiasis dietologas. Pats Stigleris rašė: „Niekas nesiruošia rekomenduoti kam nors tokią dietą (t. y. minimaliai kainuojančią), o apie tai, kad ją rekomenduoti visiems, negali būti nė kalbos“. Jis nurodė ir kitą žemos kainos dietą, sudarytą vieno dietologo 1939 metams; jos bendra metinė kaina lygi 115 dolerių. Kainos skirtumas susidarė todėl, kad tas dietologas jau rūpinosi ir maisto skanumu bei dietos įvairumu, taip pat atsižvelgė į kai kurių produktų vertingumą. Vėlesni mėginimai sudaryti optimalią dietą tiesinio programavimo metodais buvo sutikti palankiai, nes dietologų reikalavimą, kad maistas būtų aromatingas, skanus ir įvairus, analitikams pavyko išreikšti tiesiniais apribojimais. Dabar šitaip sudarinėjamas meniu daugeliui organizacijų; tokiu

---

\* Kaip tas sprendinys randamas, papasakosime 3 skyriuje.

būdu protingai taupomos lėšos, palyginus su įprastu dietologų sprendimu. Net jei uždavinį apie racionų sudarymą žmonėms laikytume sunkiai pritaikomu, vis tiek turėtume pripažinti, kad analogiškas uždavinys, susijęs su racionų

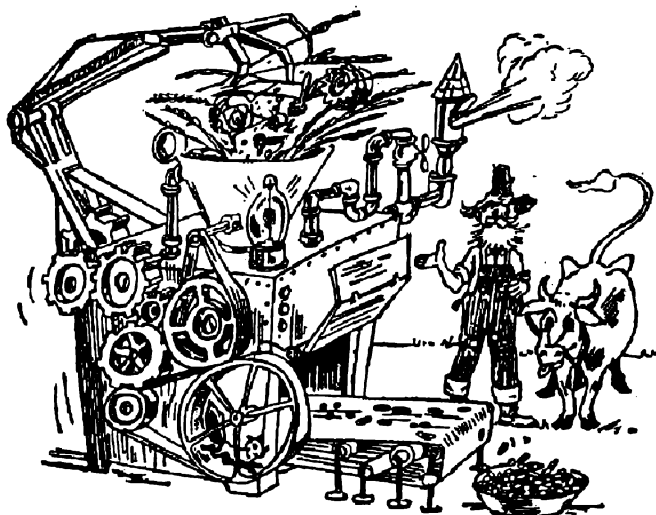


sudarymu viščiukams, galvijams arba kiaulėms, yra puikus tiesinio programavimo taikymo pavyzdys.

Uždaviniai apie dietą, arba uždaviniai apie mišinius, sprendžiami įvairiausiose srityse. Pavyzdžiui, su jais susiduriame ir sudarydami patį pigiausią pašarų mišinį gyvuliams, ir maišydami įvairius elementus, sakysime, trąšas arba chemikalus, tenkinančius bent minimalios kainos reikalavimus. Tokių uždavinių matematiniai modeliai yra to paprasto uždavinio, kurį teko spręsti šeimininkei, apibendrinimas. Netgi sudarydami pašarą gyvuliams, turime atsižvelgti į tokius apribojimus, kaip skonio savybės, tačiau šiuo atveju su jais lengviau susidoroti. Nors matematinis šių taikymų aparatas ir standartinis, kaip pavyks nustatyti visą reikalingų apribojimų sistemą, priklauso nuo to, ar analitikas sugeba įvertinti visas aplinkybes, susijusias su uždaviniu. Kartą reikėjo sudaryti pašarą gyvuliams, apribojant melasos\* kiekį, nes pašaras turėjo

\* Melasa — pašarinis sirupas, įeinantis į pašarų sudėtį.— *Rus. vert. past.*

būti gaminamas fabrike ir presuojamas rutuliukų pavidalu. Paaikškėjo, kad melasa yra labai pigi, turi gana aukštą kalcio bei proteino procentą ir karvės noriai ją ėda. Todėl optimaliame sprendinyje melasos buvo tiek, kiek tik įmanoma. Tačiau tą optimalų pašarą teko išbrokuoti. Nors ji ir pavykdavo pagaminti rutuliukų pavidalu, bet mėšlas dėl aukšto melasos procento buvo toks skystas, kad nebetiko dirvoms tręšti. Iš šito gana prozaiško pavyzdžio aiš-



kiai matyti, koks atsargus turi būti analitikas, sudarydamas praktinio uždavinio modelį. Modelio sudarymas — evoliucinis procesas, kuriame pasireiškia glaudi žmonių ir uždavinio, žmogaus ir modelio sąveika.

### **Tiekėjo uždavinys**

Taip ir turėjo anksčiau ar vėliau atsitikti: konsultantas valdymo klausimais buvo pakviestas pas pamišusį Skrybėlių\* — reikėjo jo arbatėles padaryti mažiau „pamišusias“.

\* Skrybėlius ir Zuikis Vėplys — L. Kerolio knygos „Alisa stebuklų šalyje“ veikėjai. Skrybėliaus laikrodis visada rodo šestą valandą — arbatos gėrimo metą, todėl jis su savo svečiais be paliovos geria arbatą. Pats šeimininkas kas kartą persėda į naują, gretimą, vietą prie švaraus puoduko, o svečiai pasislenka ratu paskui jį, todėl jiems tenka gerti iš nešvarių puodukų. — *Rus. vert. past.*

Jau daug metų Skrybėlius, jo draugai ir svečiai be paliovos persėdinėja iš vienos vietos į kitą, tuščiai ieškodami švarių puodukų, lėkštucių ir servetėlių, kurie jau seniai tokie purvini, kad į stalą pažvelgus net krūpteli.



— Jeigu jūs ir toliau ketinate kasdien priiminėti svečius, tai pirmiausia turite sudaryti planą,— paaikšino konsultantas Skrybėliui.

— Bet aš neturiu laiko planavimui,— skundėsi Skrybėlius,— man reikia nuolat linksminti svečius eilėraščiais, mįslėmis ir prie progos vaišinti juos arbata.

— Todėl paveskite tai man,— tarė konsultantas.— Aš kurį laiką stebėsiu, kas čia pas jus vyksta, po to jūs man truputį padėsite ir, pasinaudoję mano modeliu, pasieksime, kad arbatą gerti pasidarytų maloniau ir pigiau, negu kada nors anksčiau.

— Jau daug metų mano arbatėlės visiems yra pavyzdys. Tačiau man knieti sužinoti, kokios bus jūsų arbatėlės. Aš samdau jus,— tarė Skrybėlius.

Svečiai vėl pasistūmėjo aplink stalą, konsultantas irgi prisėdo prie jų su plunksnakočiu vienoje rankoje ir su tartimi kitoje. Ir jis, ir šeimininkas pasirašė sutartį, o jų

parašus pagal visas taisykles paliudijo Zuikis Vėplys. Sutartyje buvo nurodyta, kad konsultantas privalo visiškai ištirti tą problemą ir po mėnesio pateikti ataskaitą bei rekomendacijas.

Zemiau ir spausdiname tą ataskaitą.

## **ANALIZINIS VIDINIŲ SĄVEIKŲ TYRIMAS, ATSIZVELGIANT Į JŲ POVEIKĮ FUNKCINĖS STRUKTūros EKONOMIKAI**

Preliminarinis tiesinio programavimo modelis arbatėlių posistemii, kurį sudarė konsultantas valdymo klausimais

### **Ivadas**

Net paviršutiniškai susipažinus su arbatėlės operacijomis, vykstančiomis, vadovaujant Skrybėliui, nelieta nė mažiausių abejonių, kad čia būklė baisi. Bet koks mėginimas pritaikyti toms operacijoms panašią į analizinę metodologiją sukeltų katastrofą. Todėl rekomenduojame laikinai nutraukti arbatos gėrimo operacijas, kai tik baigsis eilinis svečių priėmimas, idant būtų galima kuo greičiau pritaikyti žemiau išdėstytą programą. Nors siūlomoji programa liečia tik vieną arbatos gėrimo operacijų aspektą, bet galime būti tikri, kad, palaipsniui įgyvendinę efektyvios kainos programą nurodomoje svarbiausioje operacijų dalyje, su didele pasisėkimo tikimybe galėsime pastūmėti pirmyn ir visų kitų racionalaus arbatėlės organizavimo problemų sprendimą. Konkrečiau kalbant, rekomenduojame sunaikinti visas turimas nešvarias servetėles ir pritaikyti naujų servetėlių pirkimo ir skalbimo programą\*. Šios bendrojo uždavinio dalies sprendimo ėmėmės, naudodamiesi galingomis priemonėmis, kurias sukūrė šiuolaikinės matematikos mokslas apie valdymą. Klausimą sprendėme šitokiu būdu.

### **Matematinis modelis**

Remdamiesi savo turtinga patirtimi matematinio modeliavimo srityje, iš visos arbatėlių sistemos sugebėjome išskirti pagrindinį posistemį, kurį galima matematiškai analizuoti. Ši analizė yra pagrįsta plačiai žinomu tiekėjo uždaviniu, kai koks nors kulinarijų gaminių tiekėjas nori apskaiciuoti, kiek servetėlių jam reikia nusipirkti ir kiek siųsti į skalbyklą, kad visą laiką turėtų jų pakankamai savo klientams. Tiekėjas stengiasi sudaryti tikslų pirkimo ir skalbimo balansą, kad minimizuotų visą servetėlių posistemo kainą. Mūsų planas — pritaikyti tą tiesinio programavimo modelį arbatėlių uždaviniui. Optimizavus servetėlių srautą, atliktąją analizę bus galima pritaikyti kitiems arbatėlės operacijų elementams. Esame įsitikinę, kad per tam tikrą laiką pavyks optimizuoti sumuštinų srautą, svečių srautą ir, galų gale, arbatos srautą.

\* Kontroliniai eksperimentai parodė, kad servetėlių būklė tokia apvertk-tina, jog jų išskalbti neįmanoma.

## Analizė

Norėdami pailiustruoti, kaip tiekėjo uždavinio modelis pritaikomas mūsų uždaviniui, iš pradžių kaupėme duomenis (pirmoji mūsų fazė) ir įsitikinome, kad vidutiniškai per savaitę jūsų arbatėlėse dalyvauja tiek svečių:

- 1) pirmadienį — 5;
- 2) antradienį — 6;
- 3) trečiadienį — 7;
- 4) ketvirtadienį — 8;
- 5) penktadienį — 7;
- 6) šeštadienį — 9;
- 7) sekmadienį — 10.

Kiekvieną dieną galima pirkti naujas servetėles po 25 centus už vieną ir laiku jas gauti (pristatymas nemokamas). Netoliese yra dvi nepriekaištingos reputacijos skalbyklos. Karaliaus skalbykloje servetėlę galima išskalbti per dvi dienas, ir tai kainuoja 15 centų, Karalienės skalbykloje tas pats darbas atliekamas per tris dienas po 10 centų už vieną. Tarkime, kad visas senąsias servetėles sudeginome, vadinasi, jų nebeturime nė vienos. Dabar sudarysime tiesinį modelį pagal jūsų duomenis.

Pirmiausia aptarsime žymėjimus. Sakykime,  $n_1, n_2, n_3, n_4, n_5, n_6, n_7$  yra atitinkamą savaitės dieną nupirktų naujų servetėlių skaičius. Analogiškai  $k_1, k_2, k_3, k_4, k_5, k_6, k_7$  ir  $q_1, q_2, q_3, q_4, q_5, q_6, q_7$  reiškia skaičių servetėlių, išsiųstų atitinkamai į Karaliaus ir Karalienės skalbyklas. Pagaliau sakykime,  $d_1, d_2, d_3, d_4, d_5, d_6, d_7$  reiškia atitinkamą dieną nepasiųstų skalbti nešvarių servetėlių skaičių. Kadangi norime minimizuoti bendrą sumą, būtiną švarių servetėlių skaičiui palaikyti, nereikia pirkti servetėlių daugiau, negu būtina tai dienai, arba siųsti servetėlę į skalbyklą, jei toliau ji nebus naudojama.

Pirmąją operacijų dieną, t. y. pirmadienį, turime pirkti tiek servetėlių, kiek jų reikia. Kadangi laukiame penkių svečių, tai

$$n_1=5.$$

Pasibaigus pirmadienio arbatėlei, galime rinktis: visas ar tik kai kurias iš penkių nešvariųjų servetėlių siųsti į greitąją Karaliaus skalbyklą, į lėtesnę Karalienės skalbyklą arba palikti jas nešvarių skalbinių dėžėje. Tai, ką darome su tomis penkiomis servetėlėmis, galime išreikšti lygtimi

$$k_1+q_1+d_1=5.$$

Bendroji visų pirmosios dienos operacijų kaina lygi

$$25n_1+15k_1+10q_1 \text{ (centų).}$$

Mūsų tikslas — nustatyti, kokias skaitines reikšmes reikia suteikti kintamiesiems (kol kas  $n_1, k_1, q_1$  ir  $d_1$  laikome kintamaisiais, iš kurių  $n_1$  lygus 5). Kai parašysime visas sprendžiamojo uždavinio lygtis, taikysime tiesinio programavimo skaičiavimo metodus ir rasime minimalios vertės sprendinį. Dabar parašysime kitus to modelio apribojimus, turėdami omenyje, kad išskalbtas servetėles gauname po dviejų arba po trijų dienų, žiūrint kuri skalbykla jas skalbė. Trečiadienio

arbatėlėje naudosisime  $k_1$  servetėlių, pasiųstų į skalbyklą pirmadienį; o  $q_1$  servetėlių bus išskalbtos ketvirtadienį. Be to,  $d_1$  servetėlių, neišsiųstų skalbti pirmadienį, galima išsiųsti kitą dieną.

Antradienio arbatėlei vėl turime nupirkti tam tikrą servetėlių skaičių, būtent,

$$n_2=6.$$

Tai, ką darome su tomis šešiomis naudotomis servetėlėmis ir su  $d_1$  nešvarių servetėlių, galime aprašyti lygtimi

$$k_2+q_2+d_2=6+d_1.$$

Ta lygtis rodo, kad susikaupė  $d_1$  nešvarių servetėlių, naudotų pirmadienį, ir šešios nešvarios servetėlės, naudotos antradienį; jas galima arba siųsti į skalbyklas, arba palikti nešvarių skalbinių dėžėje. Visų antradienio operacijų kaina lygi

$$25n_2+15k_2+10q_2 \text{ (centų).}$$

Trečiadienį iš skubiosios Karaliaus skalbyklos gaunamos švarios servetėlės, kurias vėl galima naudoti. Tos dienos arbatėlei reikia 7 servetėlių. Todėl galima arba pirkti visas septynias, arba dalį jų imti iš tų servetėlių, kurios buvo atiduotos skalbti pirmadienį. Vadinasi,

$$n_3+k_1=7.$$

Visiškai panašiai, kaip ir anksčiau,

$$k_3+q_3+d_3=7+d_2;$$

visų trečiadienio operacijų kaina lygi

$$25n_3+15k_3+10q_3 \text{ (centų).}$$

Ketvirtadienį reikalingos 8 servetėlės; jos gali būti arba naujos, arba skalbtos — tos, kurias pasiuntėme į Karaliaus skalbyklą antradienį, arba į Karalienės skalbyklą pirmadienį. Tai galima užrašyti lygtimi

$$n_4+k_2+q_1=8;$$

be to,

$$k_4+q_4+d_4=8+d_3.$$

Ketvirtadienio operacijos kainuoja

$$25n_4+15k_4+10q_4 \text{ (centų).}$$

Dabar galime iš karto užrašyti likusias tiriamojo modelio lygtis. Paprastumo dėlei tarsime, kad efektyviu arbatėlių rengimu domimės tik vieną savaitę, todėl nebeduosime skalbti servetėlių, jeigu jas galės grąžinti ne vėliau kaip sekmadienį.

Penktadienio operacijas su servetėlėmis galime išreikšti lygtimis:

$$n_5+k_3+q_2=7,$$

$$k_5+d_5=7+d_4,$$

bendroji operacijų kaina

$$25n_5 + 15k_5 \text{ (centų);}$$

šeštadienio — lygtimis

$$n_6 + k_4 + q_3 = 9,$$

$$d_6 = 9 + d_5,$$

bendroji kaina  $25n_6$  (centų);

sekmadienio — lygtimis

$$n_7 + k_5 + q_4 = 10,$$

$$d_7 = 10 + d_6,$$

bendroji operacijų kaina  $25n_7$  (centų).

Iš visų parašytųjų lygčių matome, kad reikia rasti dydžius  $n$ ,  $k$ ,  $q$  ir  $d$  (tie dydžiai yra neneigiami — arba teigiami, arba lygūs nuliui), minimizuojančius kainos funkciją  $25(n_1 + n_2 + n_3 + n_4 + n_5 + n_6 + n_7) + 15(k_1 + k_2 + k_3 + k_4 + k_5) + 10(q_1 + q_2 + q_3 + q_4)$  ir tenkinančius tiesines lygtis

$$n_1 = 5,$$

$$n_2 = 6,$$

$$n_3 + k_1 = 7,$$

$$n_4 + k_2 + q_1 = 8,$$

$$n_5 + k_3 + q_2 = 7,$$

$$n_6 + k_4 + q_3 = 9,$$

$$n_7 + k_5 + q_4 = 10,$$

$$k_1 + q_1 + d_1 = 5,$$

$$k_2 + q_2 + d_2 = 6 + d_1,$$

$$k_3 + q_3 + d_3 = 7 + d_2,$$

$$k_4 + q_4 + d_4 = 8 + d_3,$$

$$k_5 + d_5 = 7 + d_4,$$

$$d_6 = 9 + d_5,$$

$$d_7 = 10 + d_6.$$

### Optimalus sprendinys

Iš atliktų skaičiavimų išplaukia, kad, perkant ir skalbiant toliau nurodyta tvarka, švarių servetėlių užteks kiekvienam svečiui. Jų bendroji kaina bus lygi 8 doleriams 80 centų; norint aptarnauti 52 svečius, reikės nupirkti 21 naują servetėlę.

### Pirkimas ir skalbimas vienos savaltės arbatėlėms

|         |         |         |          |
|---------|---------|---------|----------|
| $n_1=5$ |         |         | $d_1=0$  |
| $n_2=6$ |         |         | $d_2=0$  |
| $n_3=7$ | $k_1=0$ |         | $d_3=0$  |
| $n_4=3$ | $k_2=0$ | $q_1=5$ | $d_4=0$  |
| $n_5=0$ | $k_3=1$ | $q_2=6$ | $d_5=2$  |
| $n_6=0$ | $k_4=3$ | $q_3=6$ | $d_6=9$  |
| $n_7=0$ | $k_5=5$ | $q_4=5$ | $d_7=10$ |
| <hr/>   |         |         |          |
| 21      | 9       | 22      | 21       |

Bendroji kaina doleriais:  $21 \times 0,25 + 9 \times 0,15 + 22 \times 0,10 = 8,80$ .

Esame įsitikinę, kad iš matematinio arbatėlių valdymo tyrimo gaunama milžiniška lėšų ekonomija, o apibendrintą anksčiau atliktą analizę bus galima pritaikyti visoms Skrybėliaus operacijoms.

### Istorinė informacija

Tiekėjo uždavinys pirmą kartą literatūroje pasirodė kaip grynai karinis uždavinys. Šis blogai užmaskuotas mėginimas nuslėpti tikrąją to modelio reikšmę ir galią buvo greitai įveiktas. Kad vaizdas būtų pilnesnis, pateiksime to uždavinio originalo formulę. Vietoj tiekėjo — karininkas, kuris operuoja lėktuvų varikliais (servetėlėmis) pagal tam tikrus reikalavimus (svečių skaičius per dieną), t. y. arba perka naujus variklius, arba siunčia remontuoti išėjusius iš rikiuotės. Remontas gali būti ir skubus (greitas aptarnavimas), ir paprastas, pigesnis (lėtas aptarnavimas). Aišku, naujas variklis kainuoja daugiau, negu vienos ar kitos rūšies remontas.

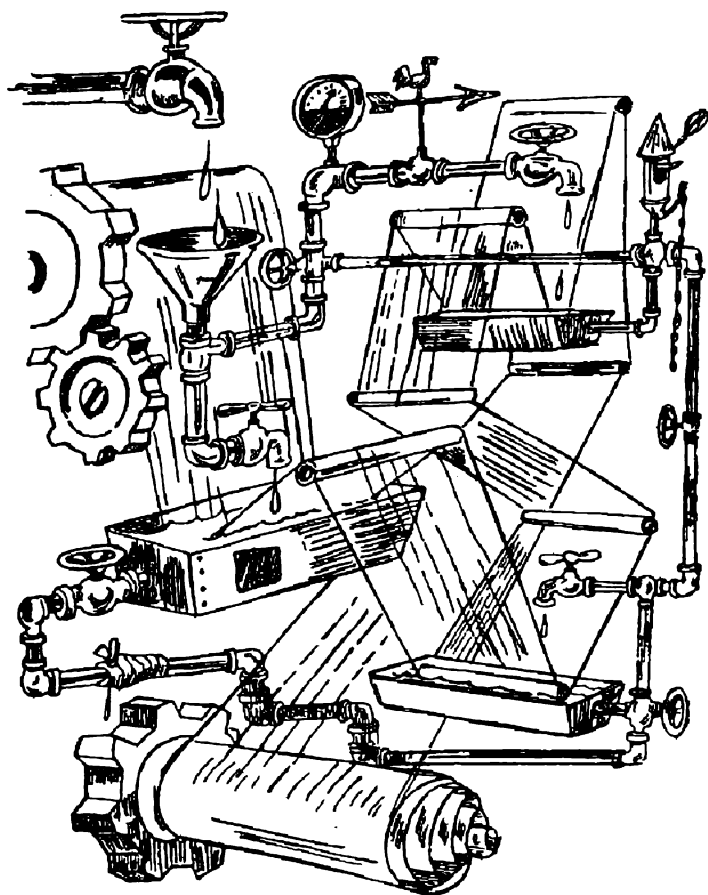
Aprašėme šį taikymo atvejį, norėdami parodyti, kad daugelio tiesinio programavimo modelių pritaikomumas įvairiems uždaviniams priklauso nuo analitiko išradingumo. Pavyzdžiui, galime įsitikinti, kad tiekėjo uždavinys iš tikrųjų yra užmaskuotas transporto uždavinys.

## Optimalaus išdėstymo uždavinys

*Tai buvo negailestingas smūgis,  
padaręs gilią žaizdą*

Paprastai kiekvienas studentas, pagal programą studijuojantis operacijų tyrimo kursą, aplanko kokią nors išsivysčiusią organizaciją (pramoninę ar kitokią) ir susipažįsta su kuria nors tos organizacijos veiklos puse. Tarp ataskaitų apie atliktą per semestrą darbą pasitaiko tokių fundamentalių darbų, kaip „Matematinio valdymo metodų pritaikymas makulatūros rinkimui“, „Apie eilių tvarkymą univermage“ arba „Vandens gatvės eismo kontrolė: transporto srauto tyrimai Venecijoje, Italija“.

Vienai studentų grupei buvo priskirta labai išsivysčiusi chemijos kompanija. Tokioje organizacijoje iškyla



daug operacinių uždavinių, susijusių su gamybos planavimu, sandėliavimu, transportavimu ir pan. Apsilankę keliose tos kompanijos įmonėse, studentai nutarė tirti operacijas, susijusias su celofano gamyba.

Medienos pulpą apdorojus šarmu ir kitomis cheminėmis medžiagomis, susidaro viskozė, kuri pro ilgus siaurus plyšius patenka į rezervuarą su sieros rūgštimi. Čia viskozė akimirksniu virsta celiulioze ir įgyja plačios juostos formą. Ta juosta patenka į kitus rezervuarus, kuriuose ji valoma, plaunama, džiovinama ant būgnų ir suvyniojama (kaip kilimas) į rulonus.

Rulonai gabenami į sandėlį ir saugomi, o, gavus užsakymų, pradedami piaustyti į siauresnes juostas (pagal gautus užsakymus). Didieji rulonai yra 60 colių pločio. Rulonas, kurį reikia supiaustyti, gabenamas į specialią patalpą ir dedamas į piaustymo mašiną. Čia mechanizmas išvynioja tą pusantro metro pločio juostą, o specialiai nustatyti peiliai tuo metu piausto celofaną į siauresnes juostas, kurios kitame mašinos gale vėl susukamos į rulonus. Piaustant gaunami ne tik užsakyto pločio rulonai, bet ir kitokie; pastaruosius stengiamasi išpiauti tokių matmenų, kad jie turėtų paklausą. Pavyzdžiui, 60 colių ruloną galima supiaustyti į tris rulonus po 15 colių, vieną 10 colių ir vieną 5 colių pločio. Jeigu 5 colių rulono niekas neužsako, tai jį tenka išmesti, dėl to susidaro nuostoliai. Vadinasi, nustačius peilius (minėtu atveju reikia keturių peilių), gaunamieji rulonai arba parduodami, arba sudaro nuostolius.

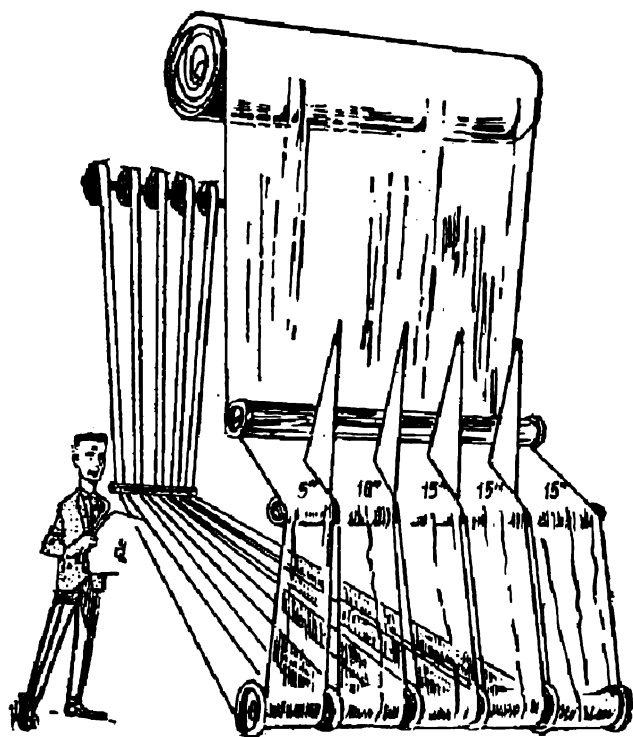
Meistras, surinkęs visos savaitės užsakymus įvairaus pločio rulonams, stengiasi juos taip sugrupuoti, kad ne tik galėtų išpiauti reikalingus rulonus, bet ir minimizuotų nuostolius. Įgudęs meistras taip sugrupuoja užsakymus, kad nuostoliai būna labai maži. Vaikščiodami po fabriką, studentai matė, kaip plėšomos ir pakuojamos ištisos atliekų krūvos. Argi negalima sutaupyti šioje nuostolingoje ir brangiai atsieinančioje operacijų dalyje? Studentai gana greitai suvokė, kad proceso valdymas faktiškai pagrįstas meistro patirtimi; jie pamėgino kiekybiškai įvertinti euristinį meistro sugebėjimą optimizuoti. Remdamasi darbu, kurį pirmą kartą atliko viena Kanados kompanija, grupė pasiūlė išmėginti modelį, jau davusį žymią ekonomiją popieriaus pramonėje. Studentai norėjo parodyti modelio pranašumą, lygindami meistro sprendinius su tie-

sinės programos sprendiniais pagal tuos pačius savaitės duomenis. Tokiomis situacijomis, kai naujamadiška matematika „kasa duobę“ žmogaus patirčiai, psichologiškai ir strategiškai labai svarbi suinteresuotos pusės parama ir bendradarbiavimas. Grupės nariams, nors jie tokiame darbe buvo naujokai, užteko nuovokos susidraugauti su meistru ir įtraukti jį į savo eksperimentą (bent jiems taip atrodė).

Meistras pateikė šitokius duomenis apie savaitės užsakymus:

| Plotis coliais | Rulonų skaičius |
|----------------|-----------------|
| 28             | 30              |
| 20             | 60              |
| 15             | 48              |

Visa tai reikėjo išpiauti iš standartinių 60 colių pločio rulonų. Buvo manoma, kad plačiųjų rulonų pakaks visiems savaitės užsakymams patenkinti. Atsižvelgiant į tą savaitę gaminamo celofano tipą, bet kurį likusį ruloną, kurio plotis mažesnis kaip 15 colių, reikėjo laikyti nuostoliu.



Vykusiai parinkus kintamuosius, pavyko sudaryti atitinkamą tiesinį modelį. Kiekvieną kartą įtvirtinus peilius, buvo gaunamas tam tikras siaurų rulonų rinkinys; reikėjo apskaičiuoti, kaip turi būti išdėstyti peiliai ir kiek didelių rulonų reikia supiaustyti, esant tai peilių padėčiai. Pavyzdžiui, norint gauti 28 colių rulonus, galima naudoti du peilius ir 60 colių ruloną supiaustyti į tris dalis — du rulonus po 28 colius ir vieną ruloną 4 colių pločio. Pirmieji du rulonai tenkintų užsakymą, o paskutinis rulonas būtų nuostolis. Vadinasi, pirmiausia reikia išsiaiškinti, kaip turi būti išdėstyti peiliai, kad gautume rulonus, kuriuos galėtume panaudoti. Kiekviena peilių padėtis yra mūsų uždavinio kintamasis, o to kintamojo reikšmė yra skaičius, rodantis, kiek 60 colių pločio rulonų reikia supiaustyti, peiliams esant toje padėtyje. Sakykime,  $x_1$  rodo, kiek 60 colių rulonų piaustoma į du 28 colių rulonus su 4 colių nuostoliu. Visos galimos peilių padėties surašytos šioje lentelėje:

| Reikiamas plotis<br>coliais | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | "Užsakytų rulonų<br>skaičius |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|------------------------------|
| 28                          | 2     | 1     | 1     | 0     | 0     | 0     | 0     | 30                           |
| 20                          | 0     | 1     | 0     | 3     | 2     | 1     | 0     | 60                           |
| 15                          | 0     | 0     | 2     | 0     | 1     | 2     | 4     | 48                           |
| Nuostoliai                  | 4     | 12    | 2     | 0     | 5     | 10    | 0     |                              |

Kintamasis  $x_3$  rodo, kiek 60 colių pločio rulonų supiaustoma į vieną 28 colių pločio ruloną ir du 15 colių rulonus su 2 colių nuostoliu. Kiti kintamieji apibrėžiami analogiškai. Kiekvienas kintamasis yra neneigiamas ( $\geq 0$ ) — arba piaustome rulonus taip išdėstytais peiliais, arba nepiaustome. Dabar jau nesunku sudaryti matematinį modelį.

Prieš rašydami lygtis, studentai išsiaiškino, kad, siekiant lankstesnio ir laisvesnio valdymo, reikiamo pločio rulonų leidžiama pagaminti daugiau, negu užsakyta. Likutį galima nugabenti į sandėlį ir panaudoti kitos savaitės užsakymams. Jeigu to nebūtų, reikėtų pagaminti lygiai trisdešimt rulonų po 28 colius, šešiasdešimt — po 20 colių ir keturiasdešimt aštuonis — po 15 colių. Šiuo atveju apribojimai leidžia pagaminti daugiau rulonų, negu reikalaujama, todėl jie reiškiami ne lygtimis, bet nelygybėmis.

Iš anksčiau pateiktos lentelės studentai parašė tris pagrindinius tiesinio modelio apribojimus — po vieną kiekvienam pločiui. Atsižvelgdami į 28 colių rulonų skaičių, jie parašė nelygybę

$$2x_1 + x_2 + x_3 \geq 30,$$

į 20 colių —

$$x_2 + 3x_4 + 2x_5 + x_6 \geq 60,$$

o į 15 colių —

$$2x_3 + x_5 + 2x_6 + 4x_7 \geq 48.$$

Be to, visi  $x$  turi būti didesni už nulį arba lygūs nuliui ( $\geq 0$ ). Pagaliau bendras nuostolių kiekis buvo pateiktas reiškiniu

$$4x_1 + 12x_2 + 2x_3 + 0 \cdot x_4 + 5x_5 + 10x_6 + 0 \cdot x_7.$$

Studentai turėjo rasti aibę  $x$  reikšmių, tenkinančių parašytąsias nelygybes ir minimizuojančių bendrąjį nuostolių matą.

Vienas sprendinys, nebūtinai minimalus, buvo tas, kuriuo naudojosi meistras. Tačiau buvo ir daugiau sprendinių. Jeigu  $x_1=15$ ,  $x_4=20$ ,  $x_7=12$ , o visos kitos  $x$  reikšmės lygios nuliui, tai nurodyti apribojimai tenkinami kaip lygybės); bendras nuostolių kiekis lygus  $4x_1 + 0 \cdot x_4 + 0 \cdot x_7 = 4 \cdot 15$  (colių). Iš to sprendinio matyti, kad 15 plačiųjų rulonų reikia supiaustyti peiliais padėtyje  $x_1$  (gaunami du 28 colių ir vienas 4 colių pločio rulonas), 20 plačiųjų rulonų — peiliais padėtyje  $x_4$  (gaunami trys 20 colių pločio rulonai) ir 12 plačiųjų rulonų — peiliais padėtyje  $x_7$  (keturi 15 colių pločio rulonai); tada bus tik 60 colių nuostolių. O gal yra sprendinių su mažesniais nuostoliais?

Atkreipsime dėmesį, kad reikšmės  $x_3=30$ ,  $x_4=20$  ir  $x_7=12$  irgi tenkintų visus apribojimus (15 colių rulonų būtų daugiau, negu reikia), o nuostolių vėl būtų 60 colių. Siūlome skaitytojui įsitikinti, kad abu tuos sprendinius atitinka nuostolių minimumas.

Suradę sprendinį, studentai drauge su meistru norėjo jį palyginti su meistro sprendiniu. Tačiau meistras pasakė, kad tų sprendinių neįmanoma lyginti, nes, sudarydami modelį, vaikinai per daug nukrypo nuo realaus uždavinio. Jis mėgino paaiškinti, kokią klaidą jie padarė.

Svarstant, kaip piaustyti ruloną, reikia atsižvelgti į jo fizines savybes. Skirtingai nuo popieriaus pramonės įmo-

nių, kuriose rulonai praktiškai gaminami be defektų, įmonėje, į kurią pateko studentai, celofano vyniojimo procesas gana netobulas. Dauguma rulonų būna panašūs į suvyniotus kilimus: viename rulono gale — įduba, kitame — iškilumas. Sunaudoti visą ruloną pavyksta labai retai. Dažniausiai tokio rulono galus tenka nupiauti, todėl toliau apdorojamas jau siauresnis rulonas. Be to, defektai dažnai aptinkami ir rulono viduje, nes vyniojamas celofanas ne visada vienodai įtempiamas. Gabalai su defektais išpiaunami panašiai, kaip ir lentų dalys, kuriose yra skylių nuo šakų. Vadinas, kur kas daugiau nuostolių atsiranda gamybos, o ne piaustymo procese.

Išsiaiškinę tuos faktus, studentai vėl ėmėsi darbo. Patikslino, kaip atsiranda nuostoliai (defektai rulonuose, brokas piaustant), ir įsitikino, kad ne visada dėl to kaltas meistras. Aptikęs rulone vietą su defektais, meistras turi prisitaikyti prie naujos situacijos ir taip pergrupuoti užsakymus, kad kuo racionaliau panaudotų nestandartinį ruloną.

Studentai aprašė savo patyrimą, įgytą, sudarant modelį, rekomenduodami geriau ištirti piaustymo procesą. Jie nurodė ir galimą problemos sprendimo būdą: piaustymo mašinoje reikia įtaisyti elektroninį prietaisą, kuris, aptikęs rulonuose defektus, perduotų duomenis skaičiavimo mašinai. Po to, remdamiesi tiesinio programavimo metodais, studentai sudarė modelį, pagal kurį galima nustatyti peilių padėtį bet kokiam rulonui, atsižvelgiant į turimus užsakymus; peiliai turi būti išdėstomi automatiškai, panaudojant skaičiavimo mašiną. Studentai preliminariai apskaičiavo būsimus nuostolius ir palygino tuo metu taikyto ir jų siūlomo metodų efektyvumus. Ataskaitos pabaigoje grupė rekomendavo būtinai palikti meistrą jo užimamose pareigose.

Pastebėta, kad žmogus, įgudęs spręsti ne per daug didelės apimties optimumo uždavinius, lengvai juos įveikia ir randa sprendinius, gana artimus optimaliems. Taip Stigleris sprendė dietos uždavinį, taip sprendė ir meistras. Kai uždavinys pasidaro sudėtingesnis, matematinio modelio vaidmuo padidėja, o žmogaus — gerokai sumažėja. Tačiau ir taip būna ne visada.

Dar palyginti neseniai naujokui būdavo pateikiama daugybė testų — pagal juos spręsdavo, kurios rūšies kariuomenėje jis bus naudingiausias. Naudojantis tais testais, buvo tiriami jo gabumai mechanikai, elektrotechnikai ir radiotechnikai, skaičiuojami balai, nagrinėjami poreikiai, ir naujokas tapdavo... pėstininku.

Dabar yra kiek kitaip. Testai vis dar naudojami, poreikiai analizuojami, bet jau yra vilties, kad naujokas užbaigs tarnybą toje vietoje, kurioje reikalingos savybės, nors šiek tiek susijusios su jo tikraisiais gabumais. Problema, kaip teisingai paskirstyti žmones pagal profesinius reikalavimus, tirama ne tik armijoje, bet ir pramonėje. Ir kaip tik tiesinis programavimas labiausiai padeda racionaliai klasifikuoti personalą. Svarbiausias tuose tyrimuose yra personalo skirstymo uždavinys, kurį galima suformuluoti ir išspręsti, naudojantis tiesinio programavimo modeliu.

Užėikime į kurį nors naujokų priėmimo centrą tuo momentu, kai ten viešpatauja tylą. Centre yra tik trys naujokai, kuriuos dėl visiškai aiškių priežasčių pavadinsime Eibliu, Beikeriu ir Čarliu\*. Jaunuoliams buvo pateikta testų serija, norint nustatyti jų gabumus radisto, programuotojo ir raštvedžio profesijoms. Gauti balai surašyti šioje lentelėje:

|          | Radistas | Programuotojas | Raštvedys |
|----------|----------|----------------|-----------|
| Eiblis   | 5        | 4              | 7         |
| Beikeris | 6        | 7              | 3         |
| Čarlis   | 8        | 11             | 2         |

Kuo aukštesnis balas, tuo didesni naujoko gabumai tai profesijai. Iš Čarlio, pavyzdžiui, išeitų tikrai geras programuotojas, bet blogas raštvedys. Centro žinioje yra trys žmonės (Eiblis, Beikeris ir Čarlis); vieną iš jų reikia pasiųsti į radistų mokyklą, kitą — į programuotojų, o trečią — į raštvedžių. Centras turi nuspręsti, ką ir kur siųsti, kad jie duotų daugiausiai naudos armijai.

\* Anglų kalboje tie vardai prasideda pirmomis abėcėlės raidėmis — Able, Beker, Charlie.— *Rus. vert. past.*

To uždavinio sprendimas, naudojantis tiesiniu programavimu ir psichologiniais testais, yra pagrįstas prielaida, kad iš eksperimentų gautais balais galima įvertinti žmogaus tinkamumą vienam ar kitam darbui (žmogaus vertės matas) ir kad bendroji viso skirstymo vertė yra lygi balų sumai. Pavyzdžiui, kai Eiblis skiriamas į pirmąjį darbą, Beikeris — į antrąjį, o Čarlis — į trečiąjį (kaip nurodyta lentelėje), bendroji balų suma yra lygi  $5+7+2=14$ . Žinoma, galima abejojti, ar teisinga taip matuoti bendrąją vertę (čia tarėme, kad priklausomybės tarp individualių skirstymų verčių yra tiesinės). Jei skaitytojui kelia nerimą naudojami psichologiniai matai, patariame paskaityti atitinkamą literatūrą.

|          | Radistas | Programuotojas | Raštvėdys |
|----------|----------|----------------|-----------|
| Eiblis   | 1        | 0              | 0         |
| Beikeris | 0        | 1              | 0         |
| Čarlis   | 0        | 0              | 1         |

Pateiktoji lentelė vadinama skirstymo lentele, nes ji išreiškia žmonių skirstymą į darbą (kiekvienas žmogus paskiriamas į vieną darbą, ir į kiekvieną darbą paskiriamas žmogus). Vadinasi, skirstymo lentelė yra kvadratinė; kiekvienoje jos eilutėje ir kiekviename stulpelyje yra po vieną vienetą. Kitas minėtų naujokų skirstymo būdas yra šitoks.

|          | Radistas | Programuotojas | Raštvėdys |
|----------|----------|----------------|-----------|
| Eiblis   | 0        | 1              | 0         |
| Beikeris | 1        | 0              | 0         |
| Čarlis   | 0        | 0              | 1         |

Šio skirstymo vertė lygi  $6+4+2=12$ . Sprendžiamajame uždavinyje ( $3 \times 3$ ) yra tik  $3 \times 2 \times 1=6$  galimi skirstymo būdai. Nesunku peržiūrėti visus galimus būdus ir įsitikinti, kad skirstymą, išreiškiamą lentele

|          | Radistas | Programuotojas | Raštvedys |
|----------|----------|----------------|-----------|
| Eiblis   | 0        | 0              | 1         |
| Beikeris | 1        | 0              | 0         |
| Čarlis   | 0        | 1              | 0         |

atitinka maksimali balų suma, lygi  $6+11+7=24$ . Kai žmonių yra labai daug, taip peržiūrėti visus atvejus praktiškai neįmanoma. Tuo tarpu tiesinį modelį galima sudaryti lengvai.

Labai naudinga būtų sprendžiamąjį uždavinį susieti su transporto uždaviniu. Iš tikrųjų čia turime keletą siuntimo punktų, vadinamų Eibliu, Beikeriu ir Čarliu. Iš kiekvieno tokio punkto į tam tikrus paskirties punktus reikia pervežti po vieną medžiagos vienetą (naująją). Kiekvienam paskirties punktui (radistas, programuotojas, raštvedys) reikia vieno iš trijų turimų vienetų. Priešingai, negu buvo daroma, sprendžiant transporto uždavinį, šiame uždavinyje turime maksimizuoti bendrąsias išlaidas (faktiškai — skirstymą atitinkančių balų sumą). Visus duomenis galime surašyti tokioje pat lentelėje, kokią buvome sudarę, sprenddami transporto uždavinį:

|          | Radistas      | Programuotojas | Raštvedys     |   |
|----------|---------------|----------------|---------------|---|
| Eiblis   | 5<br>$x_{11}$ | 4<br>$x_{12}$  | 7<br>$x_{13}$ | 1 |
| Beikeris | 6<br>$x_{21}$ | 7<br>$x_{22}$  | 3<br>$x_{23}$ | 1 |
| Čarlis   | 8<br>$x_{31}$ | 11<br>$x_{32}$ | 2<br>$x_{33}$ | 1 |
|          | 1             | 1              | 1             |   |

Kiekvienas kintamasis interpretuojamas kaip atitinkamo žmogaus paskyrimas į atitinkamą darbą; pavyzdžiui,  $x_{23}$  reiškia, kad Beikeris siunčiamas į raštvedžių mokyklą. Tie kintamieji turi būti neneigiami; faktiškai jie gali įgyti tik dvi reikšmes: 0 ir 1. Anksčiau minėtą optimalų sprendinį gauname, kai  $x_{21}=1$ ,  $x_{32}=1$ ,  $x_{13}=1$ , o visos kitos  $x$  reikšmės lygios nuliui. Prisiminę, ką darėme, sprenddami

transporto uždavinį, gauname šitokį tiesinį modelį. Reikia rasti neneigiamas kintamųjų  $x_{ij}$  reikšmes ( $\geq 0$ ), maksimizuojančias

$$5x_{11} + 4x_{12} + 7x_{13} + 6x_{21} + 7x_{22} + 3x_{23} + 8x_{31} + 11x_{32} + 2x_{33}$$

ir tenkinančias šiuos apribojimus:

$$x_{11} + x_{12} + x_{13} = 1,$$

$$x_{21} + x_{22} + x_{23} = 1,$$

$$x_{31} + x_{32} + x_{33} = 1,$$

$$x_{11} + x_{21} + x_{31} = 1,$$

$$x_{12} + x_{22} + x_{32} = 1,$$

$$x_{13} + x_{23} + x_{33} = 1.$$

Visiškai aišku, kad to uždavinio sprendinyje kintamųjų reikšmės turi būti sveikieji skaičiai (nulis arba vienas). Laimei, tiriamojo uždavinio matematinė struktūra sutampa su transporto uždavinio struktūra, todėl galime būti tikri, kad egzistuoja optimalus sprendinys, išreikštas sveikaisiais skaičiais. Kai sprendėme uždavinį apie optimalų išdėstymą, tokio reikalavimo nekėlėme. To uždavinio sprendinys galėjo būti ir toks, kad tam tikru būdu išdėstytais peiliais reikėtų piauštyti ruloną ir trupmeninę laiko dalį. Gamybos uždaviniuose nieko blogo neatsitinka, kai atsiranda ir trupmenos. Tai tik reiškia, jog gautoji kaina kiek didesnė už minimaliąją. Tačiau, kai kalbama apie žmonių skaičių, privalome rasti sprendinius, išreikštus sveikaisiais skaičiais.

Kartais skirstymo uždaviniuose reikia minimizuoti, o ne maksimizuoti tikslo funkciją. Pavyzdžiui, jeigu norėtume žmones perkelti į kitą darbą ir „testinė vertė“ būtų laikas, per kurį žmogus sugebės pakeisti profesiją, tai į naują darbą žmones taip skirstytume, kad bendrasis parengimo laikas būtų minimalus. Taikydami tuos pačius tiesinio programavimo metodus, galime išspręsti visų tipų optimumo uždavinius.

Įdomus skirstymo uždavinio atvejis (iš kurio matyti, kad tiesinis programavimas yra svarbus ir sociologijoje) yra vedybų uždavinys. Sakykime, yra 100 vyrų ir 100 moterų: norime juos taip suporuoti, kad bendroji tų porų „laimė“ būtų maksimali. Šiuo atveju reikia sudaryti skai-

čių lentelę, rodančią, kokia „laimė“ atitinka kiekvieną iš 100 porų, kurias galėtų sudaryti vyras, pasirinkdamas moterį. Jeigu  $x_{ij}$  žymėsime tą laiko dalį, kurią  $i$ -asis vyras praleidžia su  $j$ -ąja moterimi, ir jeigu mūsų tikslas — maksimizuoti bendrąją „laimę“, tai to uždavinio matematinis modelis tiesiog sutaps su personalo skirstymo uždavinio modeliu. Kaip jau minėjome, optimalieji tokių modelių sprendiniai išreiškiami sveikaisiais skaičiais (šiuo atveju 0 arba 1). Vadinasi, nors kiekvienas vyras gali padalyti savo laiką kelioms moterims, bet bendroji „laimė“ būtų didesnė, jeigu jis nebūtų mergišius. Tačiau, kaip pasakė reporteris vienam iš tiesinio programavimo kūrėjų Dž. Dancigui, pirmą kartą suformulavusiam šį uždavinį, mes būtume visiškai patenkinti ir neoptimaliais sprendiniais.

## Gamybos planavimo uždavinys

*Negirdėta, neregėta!*

*Štai kad surentė karietą:*

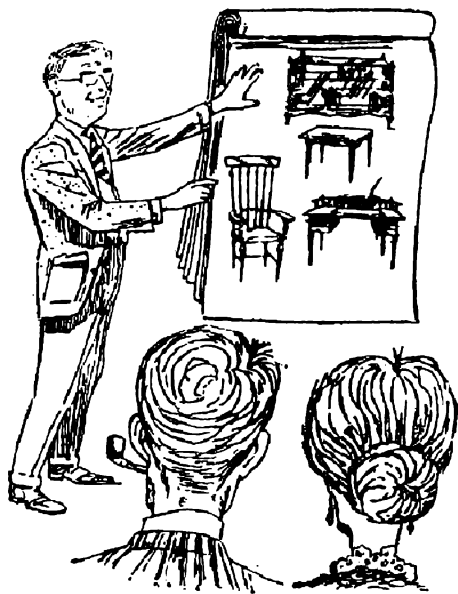
*Šimtmetį perdien nurieda.*

Dabartiniam kokios nors įmonės gamybos operacijų traktavimui būdinga tai, kad kiekviena produkcijos rūšis aprašoma, nurodant resursus vienam tos produkcijos rūšies vienetui pagaminti. Įmonės vadovas turi išspręsti šitokį uždavinį: apskaičiuoti, kiek reikia pagaminti kiekvienos rūšies vienetų (t. y. nustatyti veiklos lygį), kad kompanija gautų maksimalų pelną; be to, reikia atsižvelgti į apribojimus, susijusius su turimais resursais. Toks labai supaprastintas traktavimas (kompleksinių operacijų aprašymas pagrindinių tarpusavyje susijusių gamybos faktorių terminais) vis dėlto yra galingas įvairių gamybos procesų tyrimo metodas. Jo neatskirinama dalis yra tiesinis modelis. Norėdami susipažinti su tuo modeliu, užsukime į mėnesinį Naivuolių Baldų Kompanijos (mūsų devizas — *Proba mers facile emptorem reperit*\*) direktorių posėdį. Prezidentas Saimonas\*\* kaip tik pradeda pasakoti apie naują gamybos organizavimo būdą.

\* Gera prekė lengvai randa pirkėją (*lot.*).

\*\* Simple Simon (*angl.*) — Naivuolis Saimonas — personažas, panašus į mūsų Jonelį kvailelį. Visus vadovaujančius Naivuolių Baldų Kompanijos postus užima broliai Saimonai.— *Rus. vert. past.*

— Džentelmenai ir, žinoma, močiute Saimon! Konsultantai valdymo klausimais, ištyrę gamybos organizavimą, nurodė būdus, kaip pagerinti kompanijos veiklą. Norėdamas per tą trumpą laiką, kurį man suteikėte, paaiškinti naujo metodo esmę, leisiu sau, kiek galima, supaprastinti svarstomąjį klausimą. Daug kas iš to, apie ką pasakosiu, yra pavaizduota šiuose plakatuose.



Pirmajame plakate pavaizdavau keturis baldus, kuriuos gaminame iš gryno raudonmedžio: kėdę, stalą, rašomąjį stalą ir knygų spintą. Jūs, aišku, pastebėjote, kad, be pagrindinio ingrediento raudonmedžio, kai kurių minėtų daiktų gamybai naudojama oda; be to, reikia stiklinių durelių knygų spintai; visiems tiems baldams dar reikia klijų ir kai kurių kitų smulkių detalių, pavyzdžiui, sraigčių, į kuriuos nekreipsime dėmesio. Kai pasakiau žodį „kėdė“, visi įsivaizdavome vieną iš mūsų pasigėrėtinų gaminių, stovinčių pavyzdžių demonstravimo kambaryje. Tačiau gamybos požiūriu vaizdas yra visiškai kitoks — maždaug toks, koks pavaizduotas antrame plakate.

Jūs matote, kad kėdė iš tikrųjų yra ekvivalentiška 5 pėdoms raudonmedžio lentų, 10 žmogaus darbo valandų, 3 uncijoms klijų ir 4 kvadratinėms pėdoms odos. Mes tiesiog sujungiame tas medžiagas, kad iš jų pasidarytų kažkas panašaus į kėdę. Stalas yra ekvivalentiškas 20 pėdų lentų, 15 žmogaus darbo valandų ir 8 uncijoms klijų. Mūsų džiaugsmas ir pasididžiavimas — Saimono rašomasis stalas — tai 15 pėdų lentų, 25 žmogaus darbo valandų, 15 uncijų klijų ir 20 kvadratinė pėdų odos, o koks jis puikus! Knygų spintai reikia 22 pėdų lentų, 20 žmogaus darbo valandų, 10 uncijų klijų ir 20 kvadratinė pėdų stiklo.

Imkime vidutinę savaitės produkciją. Savaitės pradžioje viceprezidentas gamybos reikalams Sidas Saimonas gauna visų turimų resursų sąrašą, greitai apskaičiuoja ir pasako savo vaikinams, kokių ir kiek baldų reikia pagaminti. Aš visiškai neketinu Sido atleisti iš darbo, anaipol,



tik noriu duoti mokslinių patarimų, kuriais remdamasis jis galėtų padidinti pajamas. Štai šiame plakate surašyti savaitės resursai.

|                                |
|--------------------------------|
| 20 000 pėdų raudonmedžio lentų |
| 4 000 darbo valandų            |
| 2 000 uncijų klijų             |
| 3 000 kvadratinų pėdų odos     |
| 500 kvadratinų pėdų stiklo     |

Sido pareiga — nustatyti, kaip paskirstyti tuos resursus, kad gautume maksimalų pelną. Tuo tikslu jis teiraujasi viceprezidento finansiniams reikalams Hario Sai-

mono naujausių duomenų apie pelną, gaunamą iš kiekvieno produkcijos vieneto gamybos. Tuos duomenis surašiau čia:

|                               |
|-------------------------------|
| 45 doleriai už kėdę           |
| 80 dolerių už stalą           |
| 110 dolerių už rašomąjį stalą |
| 55 doleriai už knygų spintą   |

Kiek vienetų reikia pagaminti? Koku būdu galime gauti maksimalų pelną? Ar teisingai elgiasi Sidas? Leiskite parodyti, kaip mūsų naujasis tiesinio programavimo modelis atsako į šiuos ir kitus klausimus.

Man teks pasitelkti matematiką, bet stengsiuosi kuo mažiau ja naudotis. Juk visi seniai nebelankome mokyklos! Reikalingų pagaminti atitinkamos rūšies baldų skaičių žymėsiu pirmąja pavadinimo raide. Kėdžių skaičius bus  $c$ , stalų skaičius atitinkamai  $t$  ir t. t.\* Jeigu  $c=10$ , tai pagaminsiu dešimt kėdžių.

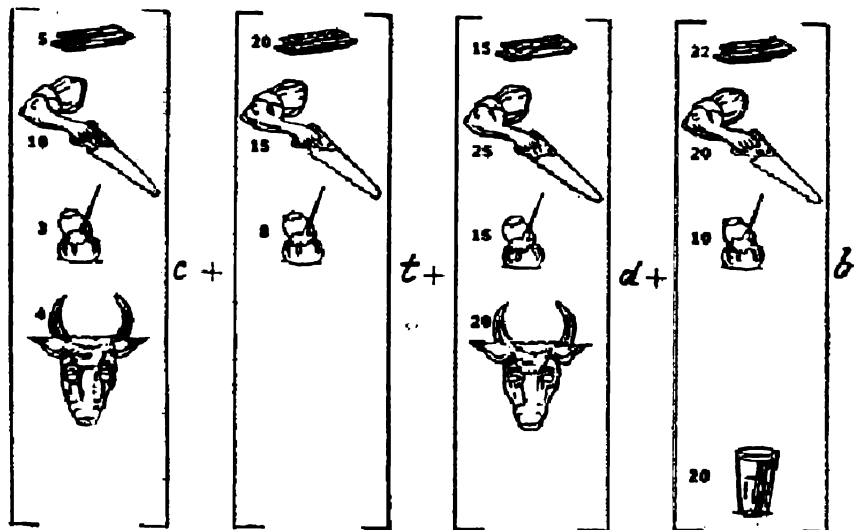
Kiekvienai kėdei sunaudojami šitokie resursai:

Jeigu pagaminsiu  $c$  kėdžių, tai tų resursų sunaudosiu  $c$  kartų daugiau:

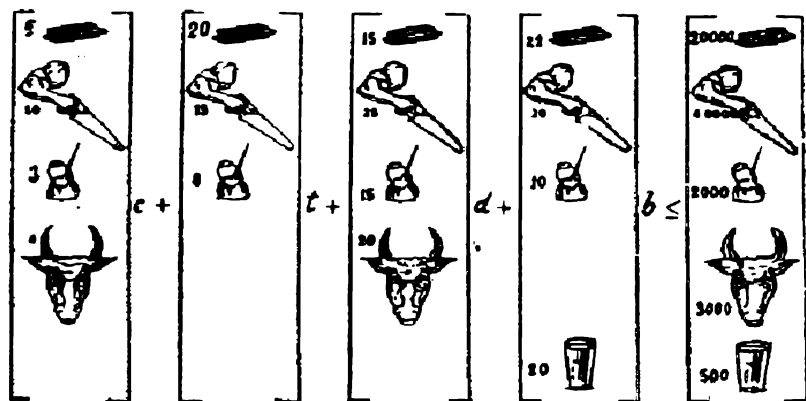


\* Angliškai kėdė — chair, stalas — table, rašomasis stalas — desk, knygų spinta — book-case. Todėl atitinkamų baldų skaičius žymimas raidėmis  $c$ ,  $t$ ,  $d$ ,  $b$ . — Rus. vert. past.

Tą patį galima pasakyti apie stalus, rašomuosius stalus ir knygų spintas. Bendrą kiekį visų resursų, sunaudotų  $c$  kėdžių,  $t$  stalų,  $d$  rašomųjų stalų ir  $b$  knygų spintų pagaminti, galiu pavaizduoti šitaip:



Tas bendras kiekis turi būti mažesnis už turimų resursų kiekį arba lygus jam; todėl gausiu štai ką:



Pelnas išreiškiamas suma

$$45c + 80t + 110d + 55b \text{ (dolerių).}$$

Dabar belieka rasti teisingas  $c$ ,  $t$ ,  $d$  ir  $b$  reikšmes, kurioms esant gaunamas maksimalus pelnas. Kaip tik tai ir stengiasi padaryti Sidas. Tiesinis programavimas laiduoja didžiausią pelną. Kas nori paklausti? Prašom, Sidai.

— Tu nepalietei daugelio klausimų. Aš įsitikinęs, kad viskas išlėks iš varžytynių. Vaikiniai, kurie čia apsilanko pravažiuodami, užsuka pas mus ir užsako baldus, bet jie nori, kad jų užsakymus išpildytume nedelsiant. O dirbdamas tokiu būdu, apie kurį čia kaip lakštingala suokei, nesu tikras, kad pagaminsiu, pavyzdžiui, nors vieną kėdę. Taip pat galiu pagaminti perdaug kėdžių arba vien tik kėdes. Kaip tu išvengsi šito?

— Labai paprastai, Sidai. Leisk man parašyti keletą nelygybių:

$$5c + 20t + 15d + 22b \leq 20\,000,$$

$$10c + 15t + 25d + 20b \leq 4000,$$

$$3c + 8t + 15d + 10b \leq 2000,$$

$$4c + 20d \leq 3000,$$

$$20b \leq 500.$$

Jeigu tau reikia pagaminti ne mažiau kaip 50 kėdžių, tai prie šio modelio užtenka pridėti vieną nelygybę

$$c \geq 50.$$

Jeigu tu nenori gaminti daugiau kaip 30 rašomųjų stalų, tai tau reikia dar prirašyti

$$d \leq 30.$$

Visa tai labai paprasta. Mūsų tiesinio programavimo modelis gali laiduoti tai ir dar daug ką. Galime čia prijungti transporto, inventoriaus ir sandėliavimo uždavinius ir pagaliau mėginti optimizuoti visą Naivuolių Sistemą. To imsimės artimiausioje ateityje. Daugiau klausimų nėra? Mano laikas baigėsi.

### 3. UŽDAVINIŲ SPRENDIMAS

(Kai kurias Euklido idėjas deriname su dvidešimtojo amžiaus intuicija, norėdami suprasti procesus, sudarančius tiesinio programavimo uždavinių sprendimo pagrindus.)

Skaitytojai, kurie domisi tik kokio nors konkretaus tiesinio programavimo uždavinio sprendimu, gali ramiai praleisti šitą skyrių ir iš karto skaityti tolesnį. Praktiškai sugebame išspręsti beveik kiekvieną tiesinio programavimo uždavinį. Yra skaičiavimo metodų, kuriais pasinaudoję, galime be mašinų spręsti nelabai grioždiškus uždavinius. Net ir tuo atveju, kai uždavinys yra sudėtingas (arba nelabai sudėtingas), tie patys skaičiavimo metodai kartu su elektroninėmis skaičiavimo mašinomis sudaro tą medžiagą, iš kurios kalamas pagrindinis analitiko ginklas. Pirmiausia, ko „mokome“ skaičiavimo mašinas, kai jos išmoksta atlikinėti pagrindinius aritmetikos veiksmus,— tai spręsti tiesinio programavimo uždavinius. Tas faktas liudija apie visur prasiskverbiančią tiesinio modelio jėgą\*.

Tikiuosi, kad tik maža skaitytojų dalis praleis pagrindines šio skyriaus mintis. Nors ir galima suprasti tiesinio programavimo esmę, gerai nežinant, kaip sprendžiamas uždavinys (ir net taikyti tas ribotas žinias praktikoje), bet vis dėlto reikia laikyti aksioma teiginį, kad tikrai galias tos srities žinias galima įgyti tik kruopščiai studijuojant teoriją, skaičiavimo metodus ir taikymą. Kadangi sąmoningai nevykdau vieno iš tų reikalavimų (ignoruoju teorinę pastato dalį), tai susidaro įspūdis, kad šios knygos „pastatas“ ir pagrindinis jos tikslas gali neišlaikyti patikrinimo, jei to klausimo neaptarsime skaičiavimo požiūriu. Kad skaitytojas giliau suvoktų dalyko esmę ir geriau jame orientuotųsi, truputį pakalbėsiu apie skaičiavimą, susijusį su tiesiniu programavimu.

Kad būtų lengviau aiškinti, vartosime standartinius matematinius žymėjimus. Nagrinėsime tik labai paprastus

---

\* Kadangi elektroninės skaičiavimo technikos raidos pradžia beveik sutampa su tiesinio programavimo atsiradimu (penktojo dešimtmečio pabaiga) ir tos dvi sritys palankiai veikė viena kitą, tai prasižėjo labai nemaloni terminų painiava. Kalbant apie programavimą, kartais neaišku, kas turima omenyje: uždavinių loginė analizė ar mašinų komandos, reikalingos uždaviniui spręsti. Pavyzdžiui, yra programa tiesinio programavimo uždaviniams spręsti.

skaitinius pavyzdžius, kurių matematiniuose modeliuose yra tik du kintamieji. Tuos kintamuosius žymėsime simboliais  $x_1$  ir  $x_2$ . Euklido planimetrijos požiūriu  $x_1$  yra vienas matmuo, arba kryptis, o  $x_2$  — kitas matmuo, arba kryptis. Kalbant matematine kalba, galima sakyti, kad tirama dvimatė Euklido erdvė. Remdamiesi geometrija, ne tik pa-vaizduosime tą erdvę dviem įprastomis koordinačių ašimis, bet ir spręsimė uždavinius (smulkiau apie tai kalbėsime vėliau).

Antrojo skyriaus pavyzdžiuose kintamieji turėjo įvairias prasmes, priklausančias nuo jų apibrėžimo konkrečiame uždavinyje. Pavyzdžiui, optimalaus išdėstymo uždavinyje iksai ( $x$ ) reiškė peilių padėtį. Kai kuriuose uždaviniuose kintamieji reiškė konkrečius pavadinimus, sakysime, raidė  $K$  dietos uždavinyje. Dabar visus kintamuosius žymėsime tik raide  $x$  su atitinkamu skaitiniu arba raidiniu indeksu. Bendruoju atveju kintamąjį žymėsime simboliu  $x_j$ . Kadangi spręsimė uždavinius, kurie yra tik skaitiniai arba simboliniai „žaisliniai“ pavyzdžiai, tai reikės priprasti prie kintamųjų be konkrečios prasmės. Pavyzdžiui, galime kalbėti apie lygtį  $x_1 + x_2 = 1$ , visiškai ne-galvodami, ką reiškia tie kintamieji.

Nors daugelyje pavyzdžių, aptariamų šiame skyriuje, yra tik du kintamieji, bet skaitytojui nebus sunku atitinkamas sąvokas ir žymėjimus pritaikyti tiems atvejams, kur reikia daugiau kintamųjų. Optimalaus išdėstymo uždavinyje buvo 7 kintamieji, bet jį formuluojant nereikėjo praplėsti savo patirties ir intuicijos, kaip, pavyzdžiui, reliatyvumo teorijoje. Erdvė, į kurią rengiamės įžengti, atvira visiems.

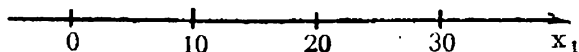
## Vienmačiai uždaviniai

(Tiems, kurie pirmą kartą susipažįsta su šiuo dalyku.)

*Atstumas — niekai; iš tikrųjų  
svarbus tik pirmasis žingsnis*

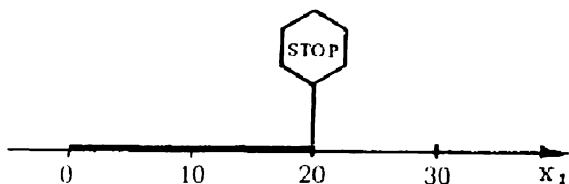
Paprasčiausias optimumo uždavinys formuluojamas šitaip: rasti maksimalią vieno kintamojo  $x_1$  reikšmę, kai nėra jokių to kintamojo apribojimų. Tokiu atveju kinta-

mojo  $x_1$  reikšmė gali būti tokia didelė, kokia tik mums atsis į galvą, todėl sakysime, kad maksimali  $x_1$  reikšmė yra neapibrėžtai didelė. Geometriškai ieškodami didžiausios kintamojo  $x_1$  reikšmės, slenkame skaičių tiese, arba ašimi.



Kadangi nėra jokių kintamojo  $x_1$  apribojimų, tai galime slinkti ta tiese kiek norint toli į dešinę, nesutikdami jokių kliūčių. Jeigu kintamąjį  $x_1$  apribosime, tai uždavinys bus įdomesnis, nors vis tiek labai lengvas. Kalbėsime apie kintamuosius, kurių reikšmės tenkina pagrindinę tiesinio programavimo sąlygą — jos negali būti neigiamos; taigi  $x_1 \geq 0$ .

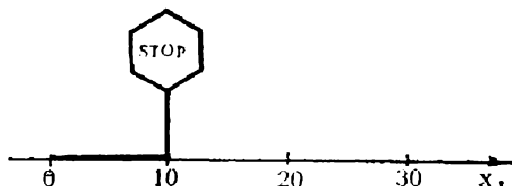
Sprendami uždavinį, kuriame reikalaujama rasti maksimalų  $x_1$ , tenkinantį nelygybę  $x_1 \leq 20$ , galime slinkti  $x_1$  ašimi nuo taško  $x_1 = 0$  iki barjero (signalu „stop“) taške  $x_1 = 20$ .



Kiekvienas tiesės taškas, esantis tarp  $x_1 = 0$  ir  $x_1 = 20$ , yra potencialus to uždavinio sprendinys. Ši taškų aibė vadinama sprendinių aibe, arba sprendinių erdve. Kadangi ieškome maksimumo, tai optimalus taškas uždavinio sprendinių erdvėje (taškas, kuriame tikslo funkcija įgyja didžiausią reikšmę) yra  $x_1 = 20$ . Jeigu ieškotume minimumo  $x_1$  reikšmės, tenkinančios nelygybę  $x_1 \leq 20$ , tai optimalus atsakymas (prisiminus neneigiamumo sąlygą) būtų  $x_1 = 0$ .

Uždavinio apie maksimumą nelygybę pakeitę nelygybe  $2x_1 \leq 20$ , laisvai judėsime  $x_1$  ašimi tik tarp  $x_1 = 0$  ir  $x_1 = 10$ ; tada optimalus sprendinys bus  $x_1 = 10$ . Šiuo būdu lengva spręsti vienmačius uždavinius su vienu apribojimu. Jeigu tas vienintelis apribojimas yra lygtis, pavyzdžiui,  $x_1 = 20$  arba  $2x_1 = 20$ , tai sprendinių erdvė sudaryta iš vienintelio taško, šiuo atveju  $x_1 = 20$  arba  $x_1 = 10$ . Tiesinio pro-

gramavimo uždaviniuose apribojimai gali būti ir lygtys, ir nelygybės, bet geometrinė tų uždavinių esmė bus aiškesnė, jei pagrindinį dėmesį skirsime nelygybėms.



Tipiškas tiesinio programavimo vienmačio uždavinio su keliais apribojimais pavyzdys yra šis uždavinys: rasti maksimalią  $x_1$  reikšmę, tenkinančią sąlygas

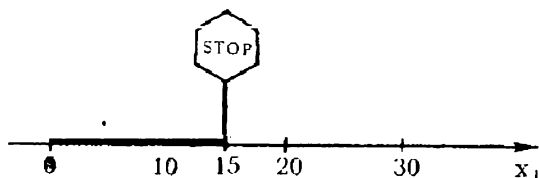
$$5x_1 \leq 75,$$

$$6x_1 \leq 30,$$

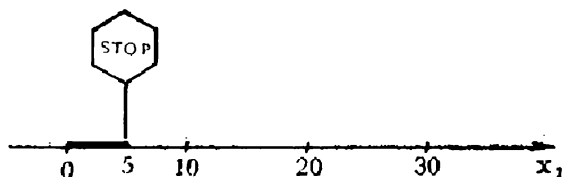
$$x_1 \leq 10$$

ir, aišku,  $x_1 \geq 0$ . Reikia rasti didžiausią  $x_1$  reikšmę, tenkinančią visas tris nelygybes. Pirmiausia išsiaiškinsime kiekvienos nelygybės sprendinių erdvę, o paskui rasime visų trijų nelygybių bendrųjų sprendinių erdvę.

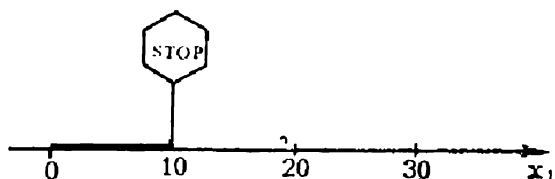
Apribojimą  $5x_1 \leq 75$  tenkina kiekviena  $x_1$  reikšmė, esanti tarp 0 ir 15.



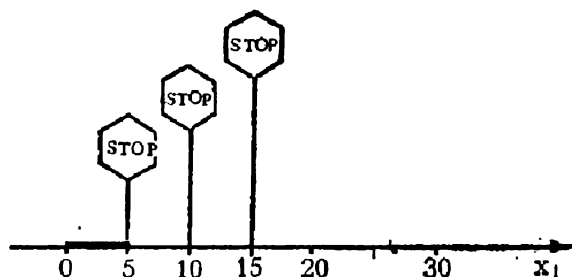
Nelygybės  $6x_1 \leq 30$  sprendinių erdvė yra tarp 0 ir 5.



Pagaliau nelygybės  $x_1 \leq 10$  sprendinių aibė yra tarp 0 ir 10.



Jeigu visus signalus „stop“ pavaizduosime viename paveiksle, tai gausime



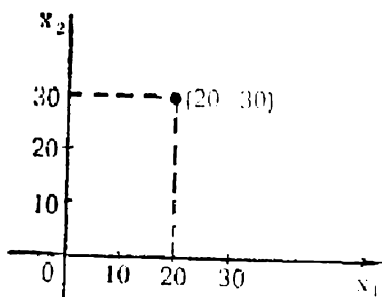
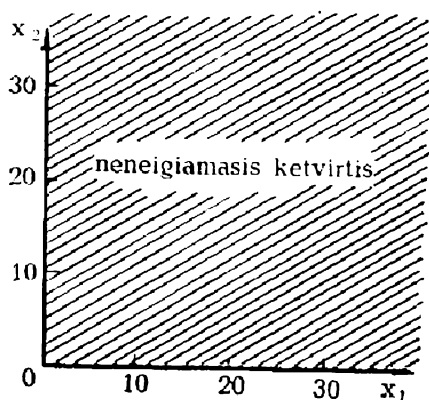
Vadinasi, judėdami nuo taško  $x_1 = 0$ , negalime peržengti taško  $x_1 = 5$ . Kiekviena reikšmė, esanti tarp 0 ir 5, tenkina visas tris nelygybes; ši taškų aibė ir yra mūsų uždavinio sprendinių aibė. Jeigu sprendinių aibėje ieškome didžiausios reikšmės, tai pradinio uždavinio su trimis nelygybėmis optimalus atsakymas yra  $x_1 = 5$ . Šitaip galime rasti bet kokio uždavinio su vienu kintamuoju optimalų sprendinį.

### Dvimačiai uždaviniai

Dabar įžengiame į areną, kurioje veikia dvimačiai uždaviniai; čia jau netrivialu ir naudinga priimti iššūkį spręsti tiesinius uždavinius. Įmdamasis tirti šią sritį, aš tikiuosi, kad skaitytojai turi pakankamą matematinį išsilavinimą. Norint suprasti toliau dėstomas mintis, būtina žinoti, kaip dvimačiai apribojimai (lygtys ir nelygybės) vaizduojami geometriškai, taip pat kiek nusimanyti apie aukštosios algebros pradmenis. Nors ir manau, kad skaitytojas tokių žinių turi, bet stengsiuosi operuoti kuo pa-

prasčiausiomis sąvokomis, kad kiekvienas galėtų įžengti į areną puikiausiai apsiginklavęs.

Du kintamuosius (arba matmenis)  $x_1$  ir  $x_2$  vaizduojame statmenomis ašimis.



Kadangi kintamųjų kitimo sritis sudaryta iš neneigiamų reikšmių, tai galima judėti ašimis tik tam tikromis kryptimis. Antai,  $x_1$  ašimi galima judėti tik į dešinę nuo taško  $O$ , koordinatų pradžios, o  $x_2$  ašimi leidžiama kilti tik aukštyn nuo koordinatų pradžios, statmenai  $x_1$  ašiai. Matematikos kalba pasakytume, kad sprendinių erdvė yra neneigiamasis ketvirtis.

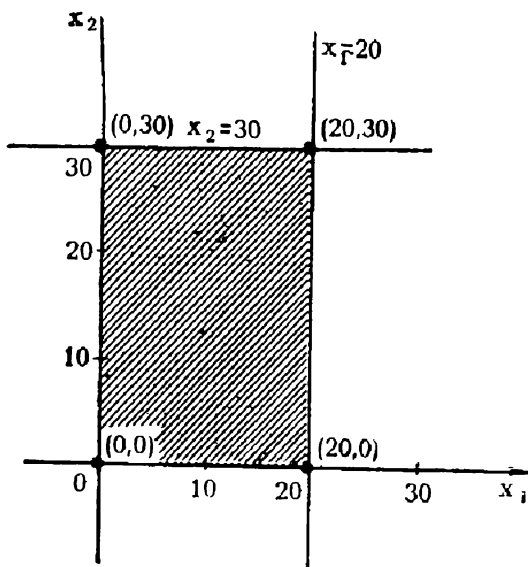
Reikšmė  $x_1=20$  nukreipia mus 20 vienetų į dešinę, o reikšmė  $x_2=30$  — 30 vienetų aukštyn. Abi tos kryptys susikerta neneigiamojo ketvirčio viduje, kaip pavaizduota paveiksle.

Vartojame sutrumpintą plokštumos taško žymėjimą:  $(x_1, x_2) = (20, 30)$ .

Kai yra tik vienas matmuo, nelygybės  $x_1 \leq 20$  sprendinių erdvė yra sritis tarp  $x_1=0$  ir  $x_1=20$ . Kiekvienas tos srities taškas tenkina sąlygą, išreikštą ta nelygybe: neneigiamas kintamasis  $x_1$  turi būti mažesnis arba lygus 20. Vadinasi, toje sprendinių erdvėje yra be galo daug sprendinių. Pavyzdžiui, skaičiai 0, 1, 2, 10,  $\sqrt{3}$  ir  $1/4$  yra leistinos kintamojo  $x_1$  reikšmės. Tačiau uždavinyje reikalaujama ne sudaryti sprendinių erdvę, bet rasti tos erdvės tašką, kuriame  $x_1$  įgyja optimalią reikšmę. Pateiktoji tikslo funkcija leidžia „apsistoti“ kuriame nors atskirame begalinės sprendinių aibės taške. Panašiai grafiškai analizuo-

jama ir sprendžiant dvimatį uždavinį. Dažniausiai pirmiausia reikia rasti begalinę aibę sprendinių, tenkinančių uždavinio apribojimus, o paskui toje aibėje parinkti tašką, optimizuojantį tikslo funkciją. Norėdami parodyti, kaip visa tai daroma, išspręsimė keletą visiškai paprastų dvimačių uždavinių.

Norime maksimizuoti  $x_1$ , laikydamiesi šių sąlygų:  $x_1 \leq 20$ ,  $x_2 \leq 30$ ,  $x_1 \geq 0$  ir  $x_2 \geq 0$ . Dvimačiame grafike nubrė-



žiame ribines tieses  $x_1=20$  ir  $x_2=30$ . Reikalingo taško ieškosime neneigiamumo sąlygų  $x_1 \geq 0$  ir  $x_2 \geq 0$  ir ribinių tiesių apribotoje užbrūkšniuotoje srityje, įskaitant ir jos kontūrą; tai ir yra sprendinių erdvė. Didžiausia kintamojo  $x_1$  reikšmė toje erdvėje lygi 20. Tačiau pastebime keistą dalyką — sprendinių erdvėje yra daug taškų, kurių  $x_1=20$ . Taškai  $(20, 0)$  ir  $(20, 30)$  yra toje erdvėje, ir abiejų  $x_1=20$ . Pasirodo, optimalus sprendinys nėra vienintelis; tiesiniame programavime tai gana įprastas atvejis, nesukeliantis papildomų rūpesčių. Iš tikrųjų čia turime be galo daug optimalių sprendinių: bet kuris atkarpos, jungiančios taškus  $(20, 0)$  ir  $(20, 30)$ , taškas yra sprendinys, nes  $x_1=20$ .

Svarbi tiesinio programavimo ypatybė yra ta, kad „geruose“ uždaviniuose (pavyzdžiui tame, kurį dabar aptariame) ribinių tiesių susikirtimo taškai yra optimalūs. Nagri-

nėjamame uždavinyje yra keturi tokie taškai — tai viršūnės  $(0, 0)$ ,  $(0, 30)$ ,  $(20, 0)$  ir  $(20, 30)$ ; maksimizuojant  $x_1$ , paskutinės dvi viršūnės yra optimalūs sprendiniai.

Jeigu pakeistume tikslo funkciją ir maksimizuotume  $x_2$ , tai optimalūs taškai būtų viršūnės  $(0, 30)$  ir  $(20, 30)$ , taip pat kiekvienas tas viršūnes jungiančios atkarpos taškas. Jeigu minimizuotume  $x_1$  arba  $x_2$ , tai pastebėtume, kad pirmoji tikslo funkcija pasiekia minimumą viršūnėse  $(0, 0)$  ir  $(0, 30)$ , o antroji — viršūnėse  $(0, 0)$  ir  $(20, 0)$ . Aišku, kad tas viršūnių poras jungiančių atkarpų taškai irgi yra atitinkamų tikslo funkcijų optimalūs sprendiniai.

Dabar pasunkinsime uždavinį, pakeitę tikslo funkciją. Ieškosime tokių  $x_1$  ir  $x_2$  reikšmių, kurios minimizuoja sumą  $x_1 + x_2$  ir tenkina tuos pačius apribojimus, kaip ir anksčiau. Ta tikslo funkcija mažiausią reikšmę įgyja taške  $(0, 0)$ ; tai yra minimumas  $x_1 + x_2 = 0$ . Tas optimalus sprendinys yra vienintelis, nes bet kuriame kitame sprendinių erdvės taške bent vienas kintamasis yra teigiamas.

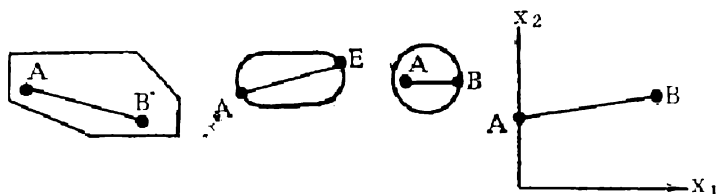
O ką galima pasakyti apie sprendinį tuo atveju, kai reikia maksimizuoti  $x_1 + x_2$ ? Nesunkiai ištyrę galimus atvejus, įsitikiname, kad vienintelis sprendinys su maksimumu  $x_1 + x_2 = 50$  yra  $(20, 30)$ .

Iš to, kas pasakyta, skaitytojas galėjo padaryti išvadą, kad bet kokios tikslo funkcijos optimalus sprendinys yra viena iš keturių viršūnių ir, gal būt, kai kurie kiti kontūro taškai. Taip iš tikrųjų būna bet kuriame „gerame“ tiesinio programavimo uždavinyje („blogas“ uždavinys yra tas, kurio tikslo funkcija nėra aprėžta; toks buvo pirmasis vienmatis uždavinys). Dabar pats laikas iškelti klausimą: ko čia, tiesą sakant, taip stengiamės? Juk jeigu tiesinio programavimo optimalus sprendinys yra kurios nors viršūnės taškas, tai užtenka rasti visas viršūnes, apskaičiuoti tikslo funkcijos reikšmę kiekvienoje viršūnėje ir pasirinkti tą, kuri optimizuoja tikslo funkciją. Pavyzdžiui, tarkime, reikia maksimizuoti funkciją  $3x_1 - 2x_2$ . Sudarome lentelę:

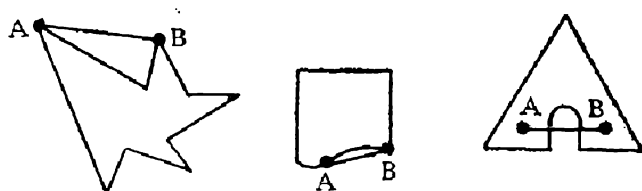
| Viršūnė    | Funkcijos $3x_1 - 2x_2$ reikšmė    | Maksimumas |
|------------|------------------------------------|------------|
| $(0, 0)$   | $3x_1 - 2x_2 = 3(0) - 2(0) = 0$    | 60         |
| $(0, 30)$  | $3x_1 - 2x_2 = 3(0) - 2(30) = -60$ |            |
| $(20, 0)$  | $3x_1 - 2x_2 = 3(20) - 2(0) = 60$  |            |
| $(20, 30)$ | $3x_1 - 2x_2 = 3(20) - 2(30) = 0$  |            |

Optimali reikšmė lygi 60, o vienintelis optimalus sprendinys — taškas  $(20, 0)$ . Skaitytojas tą patį gali padaryti su kiekviena tiesine funkcija, laikydamasis minėtų apribojimų. Visa tai atrodo labai paprasta; taip iš tikrųjų ir yra, kai nagrinėjame tokius „žaislinius“ pavyzdžius. Bendruoju atveju taip spręsti būtų sunku, nes viršūnių skaičius gali būti milžiniškas, o jų radimas — rimtas skaičiavimo uždavinys. Todėl mūsų metodas tinka tik uždaviniams, nagrinėjamiems šiame skyriuje. Tačiau net ir čia šį metodą galima pagerinti ir padaryti panašesnį į algebrinį skaičiavimo metodą (t. y. *simplekso metodą*), kuris taikomas, sprendžiant tikrus tiesinio programavimo uždavinius. Dėl tos priežasties spręsimė sudėtingesnius uždavinius, bet pirmiausia įvesime kelias pagrindines tiesinio programavimo sąvokas.

Skaitytojai, mokantys dabartinę matematiką, jau turėjo suvokti, kad dvimačio uždavinio sprendinių erdvė yra *iškilioji* taškų aibė ir kad viršūnės yra tos iškiliosios aibės *kraštiniai taškai*. Taškų aibė vadinama iškiliaja, jei bet kuriuos du jos taškus jungianti atkarpa taip pat priklauso tai aibei. Štai iškiliosios aibės:



O šitos aibės nėra iškilios:



Iškiliosios aibės taškas vadinamas kraštiniu tašku, jei jis nepriklauso nė vienai atkarpai, jungiančiai kitus du tos aibės taškus. Iškilioji aibė gali turėti ir baigtinį, ir begalinį kraštinių taškų skaičių arba visiškai neturėti to-

kių taškų. Skaitytojai patariame sugalvoti po vieną kiekvieno tipo aibės pavyzdį. Išskilioji aibė, susijusi su tiesine programa, turi baigtinį kraštinių taškų skaičių. Dažniausiai tai yra *iškilusis briaunainis*.

## Gamybos uždavinys

*Blogas tas planas, kurio negalima keisti*

Dabar nagrinėsime gamybos uždavinio, su kuriuo susidūrė Naivuolių Baldų Kompanija, supaprastintą variantą. Kadangi sprendžiame dvimačius uždavinius, tai aptarsime tik kėdžių ir stalų gamybą, kai lentų ir žmogaus darbo valandų skaičius yra ribotas; pakeisime ir turimų resursų kiekį, kad uždavinį būtų lengviau spręsti grafiškai. Todėl sprendžiamasis uždavinys yra šitoks: maksimizuoti pajamų funkciją  $45x_1 + 80x_2$  (dolerių), kai

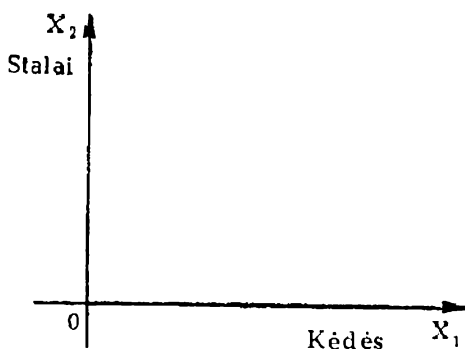
$$5x_1 + 20x_2 \leq 400,$$

$$10x_1 + 15x_2 \leq 450.$$

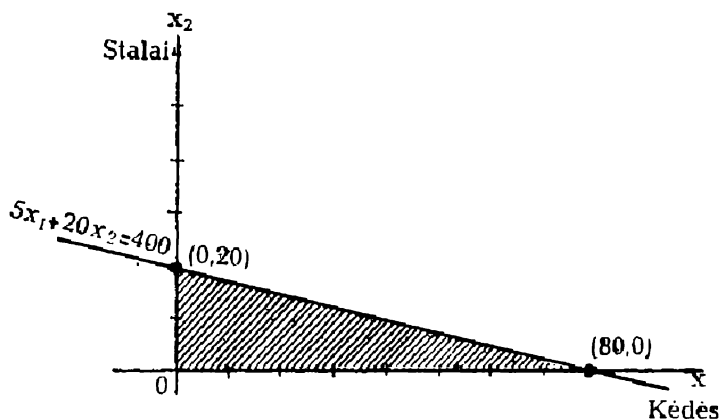
Šiuo atveju  $x_1$  reiškia gaminamų kėdžių skaičių ( $c$ ), o  $x_2$  — stalų skaičių ( $t$ ); be to,  $x_1 \geq 0$  ir  $x_2 \geq 0$ . Turime 400 pėdų raudonmedžio lentų ir 450 žmogaus darbo valandų, kurios gamybos procese turi viršti stalais ir kėdėmis.

Pirmiausia reikia sudaryti sprendinių erdvę. Iš karto pasakysime, kad ji yra pirmajame ketvirtyje.

Neneigiamajame ketvirtyje ieškosime pirmosios nelygybės  $5x_1 + 20x_2 \leq 400$  sprendinių, t. y. taškų  $(x_1, x_2)$ , kurių koordinatės tenkina šią nelygybę. Tie taškai sudaro sritį, kurią lengviausia rasti, nubrėžus ribinę tiesę  $5x_1 + 20x_2 = 400$ . Tai galima padaryti, suradus du taškus (vieną —  $x_1$  ašyje, ki-

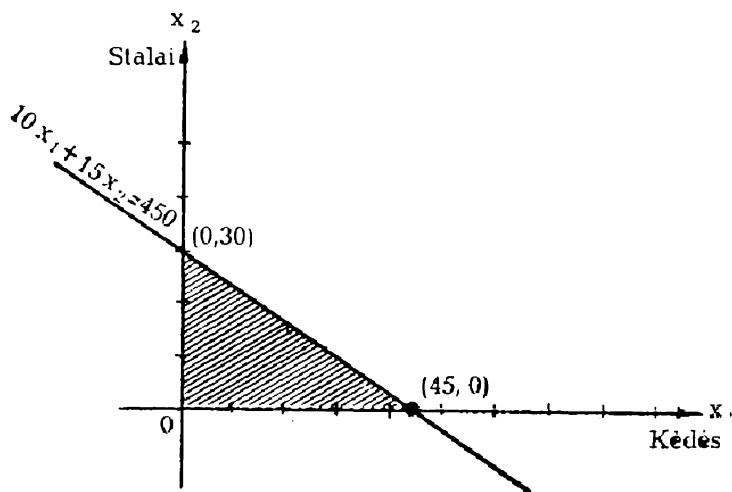


$x_1 - x_2$  ašyje), per kuriuos eina ta tiesė. Jeigu  $x_1=0$ , tai  $x_2=20$ ; jeigu  $x_2=0$ , tai  $x_1=80$ . Du taškai  $(80, 0)$  ir  $(0, 20)$  yra toje ribinėje tiesėje.



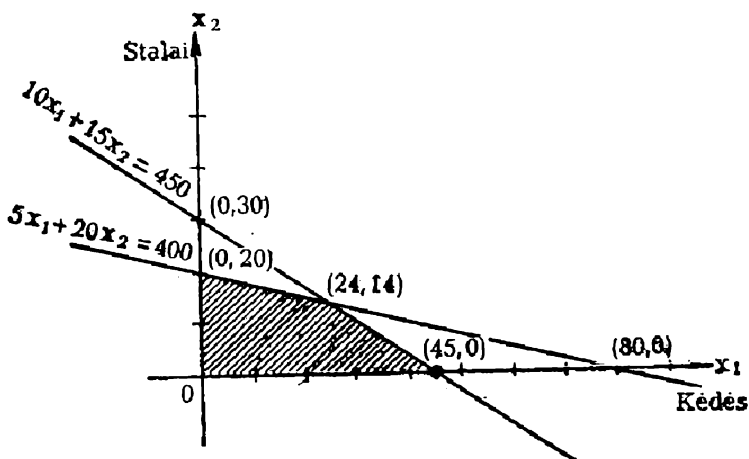
Visi užbrūkšniuotos srities ir jos kontūro taškai tenkina nelygybę  $5x_1 + 20x_2 \leq 400$ .

Antrosios nelyybės ribinė tiesė yra  $10x_1 + 15x_2 = 450$ , su koordinatų ašimis susikertanti taškuose  $(45, 0)$  ir  $(0, 30)$ ; tai pavaizduota paveiksle.



Bet kuris tos iškiliosios aibės taškas tenkina nelygybę  $10x_1 + 15x_2 \leq 450$ . Bendruosius abiejų nelygybių sprendinius

rasime, viename grafike pirmąją užbrūkšniuotą sritį uždėję ant antrosios,— bendroji jų dalis kaip tik ir yra ieškomoji sprendinių erdvė.



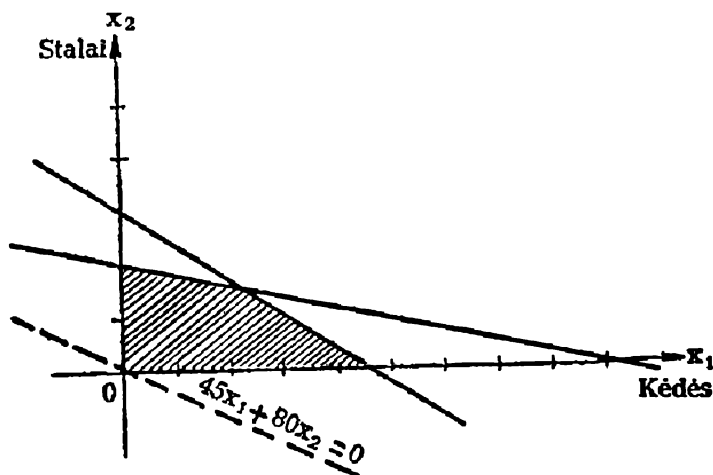
Tos bendrosios erdvės sprendiniai yra visi galimi deriniai stalų ir kėdžių, kuriuos galima pagaminti iš turimų resursų — 400 pėdų lentų ir 450 žmogaus darbo valandų. Tik taškai, priklausantys šiai sričiai, tenkina uždavinio apribojimus.

Jau žinome visus iškiliuosius sprendinių aibės kraštinius taškus, išskyrus vieną, kuriame susikerta nubrėžtosios ribinės tiesės. Šis taškas atitinka atvejį, kai sunaudojama visa žaliava. Bet kuris tiesės  $5x_1 + 20x_2 = 400$  taškas, pavyzdžiui,  $(0, 20)$  reiškia, kad sunaudojama tiksliai 400 pėdų lentų. Panašiai kiekvienas tiesės  $10x_1 + 15x_2 = 450$  taškas reiškia, kad išnaudojamos visos žmogaus darbo valandos. Pasinaudoję nelabai įmantriu algebros triuku, sužinome, kad taškas  $x_1 = 24$  ir  $x_2 = 14$  kaip tik yra abiejose tiesėse.

Uždavinys reikalauja sužinoti, kuris iš begalinės aibės taškų, esančių užbrūkšniuotoje srityje, maksimizuoja sumą  $45x_1 + 80x_2$ . Tą reiškinį galima laikyti kokios nors lygties, siejančios kintamąjį  $x_1$  su kintamuoju  $x_2$ , kairiąją pusę. Pavyzdžiui, lygtis  $45x_1 + 80x_2 = 0$  paveiksle pavaizduota brūkšnine tiese.

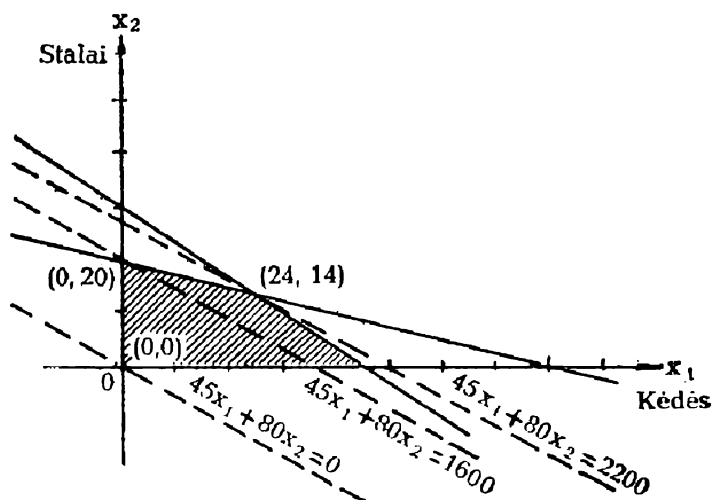
Kai Naivuolių Baldų Kompanija negamina nei stalų, nei kėdžių, iš tos tiesės matyti, kad pelnas lygus nuliui.

Pajamų tiesę taip pastumsime pajamų didėjimo kryptimi, kad būtų tenkinami visi apribojimai, susiję su turimais resursais. Tą pelno tiesę stumsime užbrūkšniuotoje sri-



tyje, kol sustosime ant kontūro (sprendinio negalima ieškoti šalia užbrūkšniuotos srities). Tokį pelno tiesės slinkimą matome žemiau pateiktame paveiksle. Pagal optimalų planą reikia pagaminti  $x_1=24$  kėdės ir  $x_2=14$  stalų, iš to bus  $45 \times 24 + 80 \times 14 = 2200$  dolerių pajamų. Pajamų funkcija „slenka“ per iškiliąją aibę kaip tik tada, kai taikome specialų algebrinį simplekso metodą.

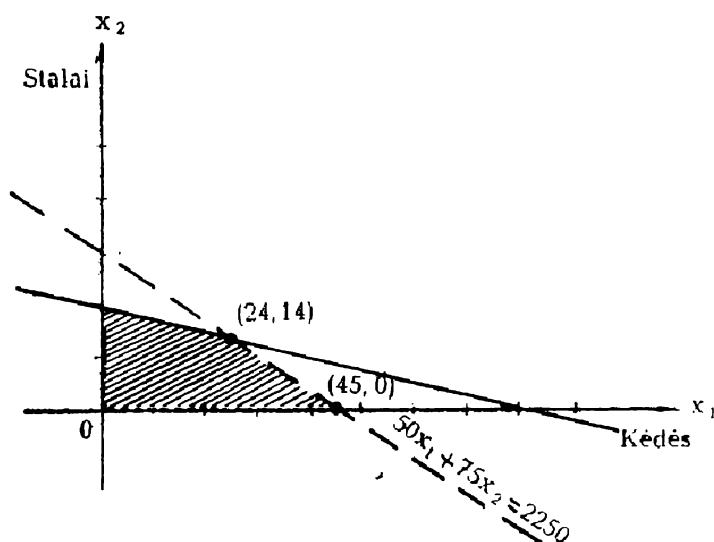
Taikydami simplekso metodą, tiriamo kurį nors kraštinį tašką, pavyzdžiui,  $x_1=0$ ,  $x_2=0$ ; įsitikinę, kad kraštinius



taškus atitinka geresni sprendiniai, tikriname kitą kraštinį tašką, atitinkantį dar geresnį sprendinį (šiuo atveju  $x_1=0$ ;  $x_2=20$ ); taip darome tol, kol pasiekiame kraštinį tašką, atitinkantį optimalų sprendinį.

Simplekso metodas yra labai ekonomiškas, nes tuo metu ieškodami geresnio gretimo taško, turime patikrinti tik nedidelį kraštinių taškų aibės poaibį. Šiuo požiūriu procesas yra labai efektyvus, o kodėl taip yra — niekas gerai nežino.

Taškas (24, 14) atitinka didelės tikslo funkcijų klasės optimalų gamybos planą. Jeigu pajamos, pardavus vieną kėdę arba stalą, keičiasi, tai keičiasi ir pajamų tiesės kryptis — ji pasisuka apie tašką (24, 14). Jeigu pajamų koeficientai gana smarkiai pakistų, tai ta tiesė taip pasvirtų, kad optimumas atsидurtų kitame taške. Kad ir kokia būtų pajamų tiesė, optimumas pasiekiamas viename iš keturių kraštinių taškų. Optimalus sprendinys nebus vienintelis tik tada, kai tikslo funkcijos grafikas sutaps su viena iškiliosios aibės kraštine. Tikslo funkcijos  $50x_1 + 75x_2$  kraštiniai taškai (24, 14) ir (45, 0), taip pat ir visi juos jungiančios atkarpos taškai bus optimalūs (žr. paveikslą).



Tokiu metodu galima spręsti bet kokią dvimatį uždavinį. Pakanka rasti taškus, reikalingus ribinėms tiesėms nubrėžti, be to, kraštinį tašką, kuriame tikslo funkcija paskutinį kartą liečia iškiląją sprendinių aibę.

Panašiai geometriškai interpretuoti galime ir daugiamčius uždavinius. Turime iškilųjį briaunainį, rastą iš pateiktųjų apribojimų, ir per tą briaunainį norime „perstumti“ tikslo funkciją optimumo kryptimi. Skaitytojas gali įsivaizduoti tą procesą, savo kambarį laikydamas iškiląja aibe, apibrėžta kokios nors tiesinės programos apribojimų. Vienas kambario kampas yra to trimačio pasaulio koordinatinių pradžia. Grindys, sienos ir lubos yra tos iškiliosios aibės, esančios neneigiamajame trimatės erdvės oktante, ribos. Bet kokia tikslo funkcija, pavyzdžiui  $x_1 + x_2 + x_3$ , optimalią reikšmę įgyja viename iš aštuonių kraštininių taškų. Atsisakius apribojimo, vaizduojamo lubomis, iškilioji aibė pasidarys neapbrėžta. Jeigu norėtume maksimizuoti  $x_3$ , kai  $x_3$  — vertikalioji kryptis, tai tikslo funkcija  $x_3$  irgi būtų neapbrėžta. Paliekame skaitytojui pačiam įsivaizduoti, kiek įvairių situacijų susidaro trimačiame pasaulyje. Dar kartą pabrėšime, kad uždaviniai su daugiau kaip dviem kintamaisiais grafiškai nesprendžiami, tačiau daugelį jų galima išspręsti simplekso metodu, naudojantis tik pieštuku ir popieriumi.

Skaitytojas, tur būt, jau pastebėjo, kad gamybos uždavinio optimalus sprendinys (24 kėdės ir 14 stalų) atitinka visus turimus resursus. Raudonmedžio kiekio apribojimas, kuris bendruoju atveju reiškiamas nelygybe  $5x_1 + 20x_2 \leq 400$ , šiuo atveju yra lygybė  $5(24) + 20(14) = 400$ . Tą patį galima pasakyti ir apie žmogaus darbo valandų skaičių:  $10 \times 24 + 15 \times 14 = 450$ . Savaime kyla klausimas: kaip pasikeis tos kompanijos pajamos, kai ji padidins kurių nors resursų kiekį? Be to, jeigu Naivuolių Kompanija iš turimų resursų sumanytų gaminti ne tik stalus ir kėdes, bet ir rašomuosius stalus, tai kaip rasti, ar pelninga dalį resursų skirti rašomųjų stalų gamybai? Tokie klausimai ir atsakymai į juos yra pagrindinės ekonominės teorijos ir gamybos valdymo problemos. Pirmasis klausimas priklauso vadinamajai marginaliajai analizei, o antrasis yra susijęs su konjunktūrinėmis išlaidomis. Turint uždavinio apie stalus ir kėdes tiesinį modelį (faktiškai bet kokį tiesinį modelį), į minėtus klausimus atsakoma, išsprendus pagalbinį uždavinį, vadinamą dualiuoju uždaviniu\*. Dualusis uždavi-

---

\* Smulkiau apie dualiuosius uždavinius ir iš dalies apie dualųjį gamybos uždavinį galite perskaityti S. Gaso knygoje «Линейное программирование (методы и приложения)». М., Физматгиз, 1961. — *Rus. vert. past.*

nys formuluojamas, remiantis pagrindinio uždavinio duomenimis. Uždavinys, dualus nagrinėtam, yra šitoks: minimizuoti resursų kainos funkciją  $400\omega_1 + 450\omega_2$ , kai

$$\begin{aligned} 5\omega_1 + 10\omega_2 &\geq 45, \\ 20\omega_1 + 15\omega_2 &\geq 80, \\ \omega_1 &\geq 0, \\ \omega_2 &\geq 0, \end{aligned}$$

čia  $\omega_1$  — nežinoma raudonmedžio resursų vieneto vertė (apskaitinė kaina), o  $\omega_2$  — nežinoma darbo jėgos resursų vieneto vertė. (Skaitytojas turėtų atkreipti dėmesį į tai, kad tikslo funkcijos koeficientai ir nelygybių dešinėsios pusės tuose uždaviniuose sukeisti vietomis. Taip pat reikia pabrėžti, kad koeficientai, pradiniam uždavinyje sudarę eilutes, dualiajame uždavinyje sudaro stulpelius.) Kadangi tie resursai yra optimalaus sprendinio „silpnosios vietos“, t. y. limituoja bendrąją produkciją, tai norėtusi tikėtis, kad dualiojo uždavinio optimaliame sprendinyje tos dualiosios kainos bus teigiamos. Pasirodo, kad optimalus sprendinys yra šitoks:  $\omega_1 = 1$ ,  $\omega_2 = 4$ , o tikslo funkcijos reikšmė lygi 2200 dolerių. Tą faktą aiškiname šitaip.

Padidinus vienetu raudonmedžio kiekį, gaunamos papildomos pajamos, lygios 1; padidinus vienetu darbo jėgą, jos lygios 4. Jeigu kurių nors resursų, pavyzdžiui raudonmedžio, yra perteklius, tai, toliau didinant jų kiekį, pajamos nedidės; vadinasi, jo „papildomas pelningumas“ (marginalioji reikšmė) bus lygus nuliui. Pagrindinė tiesinio programavimo teorema (dualumo teorema) tvirtina: jeigu pradinis uždavinys turi baigtinį optimalų sprendinį, tai tokį sprendinį turi ir dualusis uždavinys, be to, abiejų tikslo funkcijų optimalios reikšmės sutampa. Taigi įsidėmėsimė, kad visų resursų minimalioji vertė yra lygi bendroms pajamoms (2200 dolerių).

Kadangi pagal pradinio uždavinio optimalų sprendinį reikia gaminti ir kėdes, ir stalus, tai konjunktūrinės išlaidos lygios nuliui. Tai atspindi ir dualiojo uždavinio optimalus sprendinys, nes abu apribojimai tampa lygybėmis, kai  $\omega_1 = 1$ ,  $\omega_2 = 4$ . Jeigu pagal optimalų sprendinį reikėtų gaminti, pavyzdžiui, vien tik stalus, tai konjunktūrinės išlaidos kėdėms, lygios pirmosios nelygybės kairiosios ir dešinėsios pusės skirtumui  $5\omega_1 + 10\omega_2 - 45$ , būtų teigiamos, t. y. kėdės gamybai reikalingos žaliavos kaina būtų

didesnė už pajamas, gaunamas, pardavus vieną kėdę. Vadinasi, išlaidos kėdei gaminti būtų didesnės už išlaidas, susidarancias iš visų tų resursų, gaminant stalus. Visiškai panašiai analizuojama, norint išsiaiškinti, ar verta gaminti trečio pavadinimo produkciją, pavyzdžiui, rašomuosius stalus. Skaitytojui siūlome grafiškai išspręsti uždavinį, dualų uždaviniui apie stalus ir kėdes, ir atsakyti į šį klausimą: ar verta Naivuolių Baldų Kompanijai gaminti rašomuosius stalus, jei pajamos, gaunamos, pardavus tokį stalą, lygios 110 dolerių, o jo gamybai sunaudojama 15 vienetų raudonmedžio ir 25 vienetai darbo jėgos?

Remiantis tiesiniu programavimu, galima atsakyti į anksčiau minėtus klausimus, nes labai dažnai praktikoje daug svarbiau žinoti konjunktyrines išlaidas ir papildomą pelningumą, negu patį optimalų sprendinį.

### **Dietos uždavinys (dar kartą)**

*Kukliai rėdykis ir prie stalo lik saikingas;  
Tik pabučiuok mane — ir būk laimingas.*

Norėdami pavaizduoti geometrinę situaciją, kuri truputį skiriasi nuo ką tik aptartosios, grįšime prie pusryčių ruošimo uždavinio, kurį sprendė šeimininkė. Vietoj raidės  $K$  rašysime  $x_1$ , o vietoj  $C$  — raidę  $x_2$ . Matematinio požūriu reikia rasti neneigiamas  $x_1$  ir  $x_2$  reikšmes, minimizuojančias bendrąją kainą

$$3,8x_1 + 4,2x_2$$

ir tenkinančias sąlygas

$$0,10x_1 + 0,25x_2 \geq 1,$$

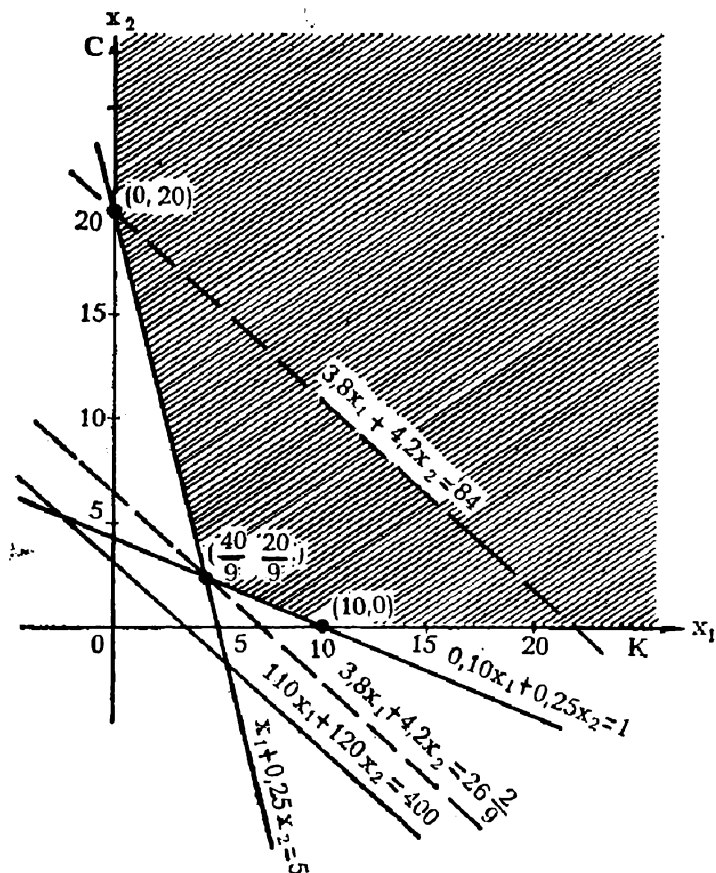
$$1,00x_1 + 0,25x_2 \geq 5,$$

$$110,00x_1 + 120,00x_2 \geq 400.$$

Kaip ir anksčiau, nubrėžę ribines tieses, gauname žemiau pavaizduotą užbrūkšniuotą sprendinių erdvę. Ta iškilioji aibė nėra iškilusis briaunainis ir nėra aprėžta nei  $x_1$ , nei  $x_2$  kryptimi.

Iš grafiko matyti, kad kalorijų kiekis yra pakankamas, kai tenkinami kiti apribojimai. Ribinė kalorijų tiesė  $110x_1 +$

$+120x_2=400$  nėra iškiliųjų sprendinių aibės kontūro dalis. Šiuo atveju ji nereikalinga, todėl toliau į kalorijų tiesę nekreipsime dėmesio. Paveiksle brūkšnine linija pa-



vaizdavome dvi kainos tieses:  $3,8x_1 + 4,2x_2 = 84$  (centų) ir minimalios kainos tiesę  $3,8x_1 + 4,2x_2 = 26 \frac{2}{9}$  (cento). Optimalus sprendinys yra vienintelis; pagal jį šeimininkė turi nupirkti  $x_1 = \frac{40}{9}$  uncijų  $K$  ir  $x_2 = \frac{20}{9}$  uncijų  $C$ .

Tam uždaviniui dualus yra šitoks uždavinys: rasti ne-  
neigiamas kintamųjų  $w_1$ ,  $w_2$  ir  $w_3$  reikšmes, maksimizuojančias reiškinių

$$w_1 + 5w_2 + 400w_3$$

ir tenkinančias sąlygas

$$0,10w_1 + 1,00w_2 + 110,00w_3 \leq 3,8,$$

$$0,25w_1 + 0,25w_2 + 120,00w_3 \leq 4,2.$$

Šiuo atveju kintamąjį  $w_i$  galima laikyti  $i$ -osios maisto medžiagos vieneto kaina. Vadinas, sprendžiant dualųjį uždavinį, ieškoma tokių  $w_i$  reikšmių, kurioms esant minimalios dietos kaina yra maksimali; be to, tariama, kad kiekvienos rūšies vienetė esančių maistingųjų medžiagų vertė yra ne didesnė už to vieneto kainą.

Tiesiniam programavimui pritaikius geometriją, begalinę sprendinių aibę galima laikyti erdvės, kurios matmenų skaičius atitinkamai parinktas, taškų aibę. Sprendinių aibė yra iškilioji, todėl galima tirti baigtinę, nors dažniausiai gana didelę, kraštinių taškų aibę (išskyrus tą atvejį, kai maksimumas yra be galo didelis). Tiesinio programavimo žavesį ir jėgą lemia sąveika vaisingų skaičiavimo metodų ir skaitingų svarbių jo taikymų — tai yra neatskiriama.

---

#### 4. TIESINIS POPURI

(Rinkiny s iš Tiesinio Programavimo Salies laukų)

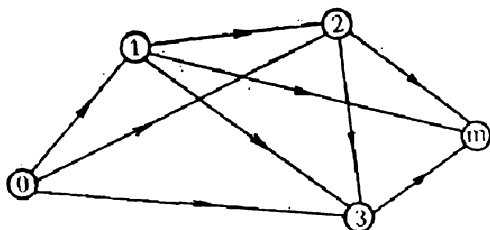
Bet koks mėginimas apžvelgti visus tuos objektus, uždavinius ir taikymus, su kuriais galima sieti tiesinį programavimą, būtų nesėkmingas. Tos sritys yra pernelyg plačios ir įvairios, kad jas būtų įmanoma aprėpti vienoje knygoje. Ankstesniuose skyriuose pasakojau apie svarbiausius, nors ir standartinius, Tiesinio Programavimo Salies uždavinius. Siame skyriuje toliau aptarinėsime ir papildomus taikymus, ir kai kuriuos neįprastus klausimus, susijusius su tiesiniu programavimu.

## Tinklų uždaviniai

*Atsipeikėk, o Sent Džonel Pasitrauk nuo maišaties,  
Nes karališką puikybę pažaboti tau reikės.  
Jeigu nieko iš lemties nebelaukiam po šia saule,  
Apsidairom — ir eime į geresnį gal pasaulį.  
Pasižvalgom — nuostabul — po žmonijos balaganą:  
Tas didžiulis labirintas, pripažinkim, ne be plano.*

Gana svarbią tiesinio programavimo uždavinių klasę sudaro tinklų uždaviniai. Panagrinėsime kelis nesudėtingus pavyzdžius.

Sakykime, naudojames koku nors transporto tinklu (vamzdynu, geležinkelių ar telefono sistema). Iš vieno to tinklo taško, vadinamo siuntimo punktu, reikia pasiųsti

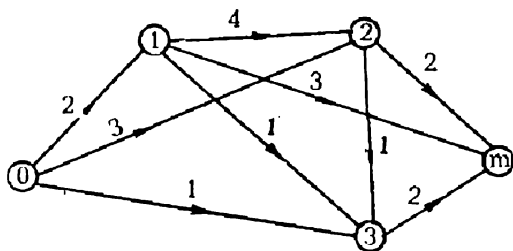


vienarūšius vienetus (naftą, mašinas, pranešimus) į tam tikrą vietą, vadinamą paskirties punktu. Be tų dviejų punktų, tinkle yra daugybė tarpinių mazginių punktų, kuriuos keliai jungia vieną su kitu ir su dviem minėtais punktais. Tuos mazginius punktus galima interpretuoti kaip vietas, kuriose yra perėjos iš vieno kelio į kitą. Siuntimo punktą žymėsime 0, paskirties punktą —  $m$ , o mazginius punktus — skaičiais arba raidėmis. Štai tipiškas nedidelio tinklo pavyzdys, kurį toliau aptarinėsime.

Kelią, jungiantį mazginius punktus 2 ir 3, žymėsime simboliu  $(2, 3)$ , o bendroju atveju — simboliu  $(i, j)$ . Paveiksle pavaizduoto tinklo keliai yra nurodytos krypties, t. y. krovinys transportuojamas rodykle nurodyta kryptimi. Jeigu krovinys vežamas dviem kryptimis, tai atitinkami mazgai sujungiami dviem priešingų krypčių keliais.

Kiekvienu keliu transportuojamas tik baigtinis krovinys. Pavyzdžiui, per valandą vamzdžiu gali pratekėti tik tam tikras naftos kiekis, ryšio linija per dieną gali būti naudojama tik tam tikrą skaičių kartų ir t. t. Vadinas, kiekvienas kelias yra susijęs su krovinio srauto viršutiniu

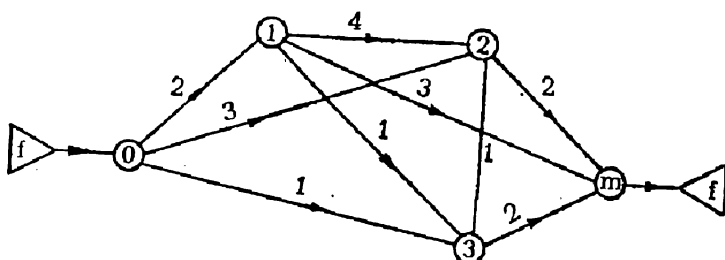
rėžiu — to kelio laidumu. Maksimalų srautą, pavyzdžiui, keliu (2, 3), žymėsime simboliu  $f_{23}$ . Čia pateiktame paveiksle tokie viršutiniai rėžiai yra nurodyti kiekvienam keliui; šiuo atveju  $f_{23}=1$ .



Kokio nors krovinio srautas iš punkto 0 vienais keliais patenka į tarpinius mazginius punktus, po to kitais keliais — į kitus mazginius punktus arba į paskirties punktą; taip vyksta tol, kol visas krovinys iš punkto 0 patenka į punktą  $m$ . Kitaip sakant, tinklas turi tenkinti *srauto nenutrūkstamumo* tarpiniuose mazguose sąlygą: tai, kas patenka į mazgą, turi ir išeiti iš jo. Ta prielaida atitinka Kirchhofo dėsnių elektros grandinei.

### Maksimalaus srauto uždavinys

Pirmasis uždavinys, kurį dabar nagrinėsime, — maksimalaus srauto uždavinys: pasiūsti kuo didesnę krovinių iš punkto 0 į punktą  $m$ . Tarkime, punkte 0 yra neribotas krovinys ir vienintelė kliūtis tam kroviniui pervežti į punktą  $m$  yra ribotas kelių laidumas, išreiškiamas nurodytais viršutiniais rėžiais. Vadinas, norime rasti galimą maksimalų srautą (žymėsime jį raide  $f$ ) krovinio, transportuojamo iš siuntimo punkto į paskirties punktą nurodyto lai-



dumo keliais. Krovinio srautas  $f$  turi praeiti šiuo tinklu taip, kaip nurodyta paveiksle.

Sudaryti matematinį modelį tokiam tinklui nėra sunku. Sakykime,

1) kintamasis  $x_{ij}$  yra krovinys, pristatytas iš mazginio punkto  $i$  į mazginį punktą  $j$ ,  
ir prisiminkime, kad

2)  $f$  — bendras tinklo srautas,

3)  $f_{ij}$  — srauto keliu  $(i, j)$  viršutinis rėžis,

4) srautas tarpiniuose mazguose nenutrūksta,

5) srautas  $f$  ir kintamieji  $x_{ij}$  yra neneigiami.

Dabar galime suformuluoti šitokį tiesinio programavimo uždavinį: reikia maksimizuoti  $f$ , kai

$$x_{01} + x_{02} + x_{03} = f,$$

$$-x_{01} + x_{12} + x_{13} + x_{1m} = 0,$$

$$-x_{02} - x_{12} + x_{23} + x_{2m} = 0,$$

$$-x_{03} - x_{13} - x_{23} + x_{3m} = 0,$$

$$-x_{1m} - x_{2m} - x_{3m} = -f,$$

$$x_{01} \leq 2,$$

$$x_{02} \leq 3,$$

$$x_{03} \leq 1,$$

$$x_{12} \leq 4,$$

$$x_{13} \leq 1,$$

$$x_{1m} \leq 3,$$

$$x_{23} \leq 1,$$

$$x_{2m} \leq 2,$$

$$x_{3m} \leq 2;$$

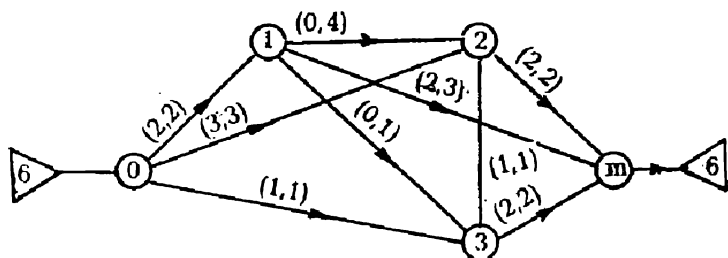
čia  $x_{ij} \geq 0$  ir  $f \geq 0$ .

Pirmoji lygtis rodo, kad srautas  $f$ , patenkantis į punktą 0, yra lygus sumai srautų, nukreiptų iš punkto 0 į kitus mazginius punktus. Analogiškai interpretuojamos ir kitos lygtys. Kiekvieną mazginį punktą atitinka viena lygtis. Nelygybės yra srautų atitinkamais keliais apribojimai iš viršaus.

Uždavinys, aišku, išsprendžiamas, nes visi apribojimai tenkinami, kai  $x_{ij}=0$  ir  $f=0$ . Tačiau tai yra, kaip sakoma, trivialus sprendinys.

Nors maksimalaus srauto uždavinį galima išspręsti simplekso metodu, bet matematinė šių uždavinių struktūra yra tokia, kad galima taikyti ir kitus metodus, kuriais greičiau ir lengviau pasiekiamas tikslas. Net labai didelės apimties uždaviniai su tūkstančiais mazginių punktų yra visiškai įveikiami. Tačiau mes nesigilinsime į tų metodų esmę, tik paminėsime keletą juos liečiančių faktų.

Pateiktasis tinklas yra paprastas, todėl lengva pastebėti, kad maksimalus srautas  $f=6$ . Kitame paveiksle prie kiekvieno kelio  $(i, j)$  parašėme skaičių porą, pavyzdžiui  $(0, 4)$ , kurios pirmasis skaičius  $x_{ij}$  lygus srautui tuo keliu, o antrasis  $f_{ij}$  — srauto tuo keliu viršutiniam rėžiui. Suprantama, kad  $x_{ij} \leq f_{ij}$ . Optimalus sprendinys pavaizduotas šiame paveiksle:



Skaitytojas mato, kad keliai, išeinantys iš punkto 0, yra užpildyti — atitinkamų srautų padidinti neįmanoma. Jeigu tuos kelius —  $(0, 1)$ ,  $(0, 2)$  ir  $(0, 3)$  — galėtume pašalinti, tai siuntimo punktą atkirstume nuo paskirties punkto. Tokių kelių aibę vadinsime *piūviu*. Piūvis — tai tokia kelių aibė, kurią pašalinus tinklas padalijamas į dvi tarpusavyje nesusijusias dalis; vienoje iš jų turi būti siuntimo punktas, o kitoje — paskirties. Keliai, priklausantys vienam piūviui, turi tam tikrą bendrą laidumą. Nagrinėjamu atveju jis lygus 6. Piūvį sudarančių kelių  $(0, 3)$ ,  $(1, 3)$ ,  $(1, m)$ ,  $(2, 3)$  ir  $(2, m)$  bendras laidumas lygus 8. Pagrindinė tinklų teorijos teorema teigia: maksimalus srautas bet kokiame tinklu iš punkto 0 į punktą  $m$  yra lygus laidumui mažiausiai laidaus piūvio, esančio tarp punktų 0 ir  $m$ . Minutę pagalvojęs, kiekvienas pasakys, kad ta

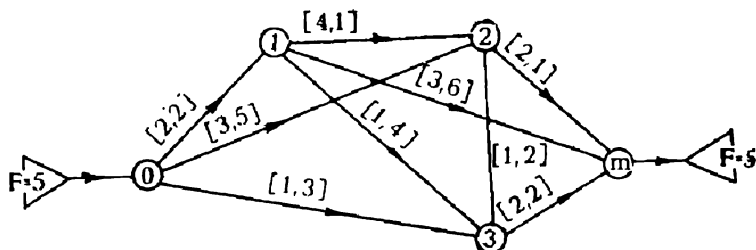
teorema intuityviai aiški. Algebrinis jos įrodymas užimtų šiek tiek daugiau laiko. Efektyvūs skaičiavimo metodai maksimalaus srauto uždaviniui spręsti remiasi minėtosios teoremos rezultatais. Be to, tie metodai laiduoja, kad kintamieji (bendrasis srautas ir srautas kiekvienu keliu) bus išreikšti sveikaisiais skaičiais.

### Pigiausio srauto uždavinys

Kitą svarbią tinklų uždavinių klasę sudaro pigiausio srauto uždaviniai. Toliau pamatysime, kad šiai uždavinių klasei priklauso ir transporto uždavinys, ir trumpiausio kelio uždavinys.

Pigiausio srauto, kaip ir maksimalaus srauto uždavinyje yra duotas koks nors kelių tinklas, kuriuo iš siuntimo į paskirties punktą reikia pristatyti vienerūšio krovinio vienetus. Krovinio vieneto transportavimo iš punkto  $i$  į punktą  $j$  kaina  $c_{ij}$  susijusi su kiekvienu keliu  $(i, j)$ . Tam tikrą krovinio vienetų kiekį  $F$  iš siuntimo punkto reikia taip nugabenti į paskirties punktą, kad bendroji transportavimo kaina būtų minimali. Tarkime, mazguose tenkinama srauto nenutrūkstamumo sąlyga ir srautas  $x_{ij}$  keliu  $(i, j)$  yra neneigiamas ir aprėžtas, t. y.  $0 \leq x_{ij} \leq f_{ij}$ . Matematinis to uždavinio modelis yra analogiškas maksimalaus srauto uždavinio modeliui, todėl jį pailiuosime, naudodamiesi ankstesniame skyrelyje aprašytu tinklu. Atkreipsime dėmesį, kad bendrasis srautas  $F$  dabar yra žinomas (maksimalaus srauto uždavinyje reikėjo rasti bendrąjį srautą  $f$ ).

Sakykime,  $F=5$  (jis negali būti didesnis kaip 6). Krovinio vieneto transportavimo kainą  $c_{ij}$  keliu  $(i, j)$  ir srauto tuo keliu viršutinį rėžį  $f_{ij}$  paveiksle nurodysime skaičių pora  $[f_{ij}, c_{ij}]$ . Aptariamuoju atveju turime



Pavyzdžiui, prie kelio (2, 3) parašyta  $f_{23}=1$ ,  $c_{23}=2$ .

Matematinis modelis yra šitoks: reikia minimizuoti

$2x_{01}+5x_{02}+3x_{03}+x_{12}+4x_{13}+6x_{1m}+2x_{23}+x_{2m}+2x_{3m}$ ,  
kai

$$\begin{array}{rcl} x_{01}+x_{02}+x_{03} & & =5, \\ -x_{01} & +x_{12}+x_{13}+x_{1m} & =0, \\ -x_{02} & -x_{12} & +x_{23}+x_{2m}-x_{3m}=0, \\ & -x_{03} & -x_{13} & -x_{23} & -x_{3m}=0, \\ & & -x_{1m} & -x_{2m} & =-5, \\ x_{01} & & & & \leq 2, \\ & x_{02} & & & \leq 3, \\ & & x_{03} & & \leq 1, \\ & & & x_{12} & \leq 4, \\ & & & & x_{13} & \leq 1, \\ & & & & & x_{1m} & \leq 3, \\ & & & & & & x_{23} & \leq 1, \\ & & & & & & & x_{2m} & \leq 2, \\ & & & & & & & & x_{3m} \leq 2, \end{array}$$

be to, visi  $x_{ij} \geq 0$ .

Nekalbėsime apie specialų metodą tokiems uždaviniams spręsti, o tiesiog pasakysime, kad, jį taikant, labai efektyviai sprendžiami gana grioždiški uždaviniai. Tačiau čia galime išnagrinėti labai įdomų to uždavinio atvejį, kuriam spręsti yra gana paprastas metodas.

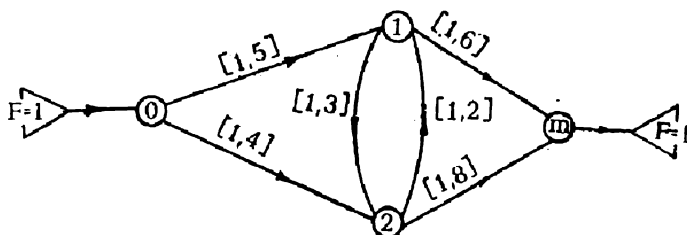
### Trumpiausio kello uždavinys

Įsivaizduokime, kad nagrinėjamas tinklas — tai kelių žemėlapis, siuntimo punktas — miestas, iš kurio išvykstame, paskirties punktas — miestas, į kurį važiuojame, o  $c_{ij}$  — atstumas tarp miestų (tarpinių punktų). Tada, turė, kad  $F=1$  ir kad viršutiniai rėžiai  $f_{ij}=1$ , iš ankstesniojo užda-

vinio galime suformuluoti šitokį: rasti minimalų atstumą tarp siuntimo ir paskirties punktų. Iš siuntimo punkto į paskirties punktą taip siunčiame vieną vienetą (srautą), kad bendrasis atstumas, kurį nukeliaus tas vienetas, būtų minimalus. Tokį uždavinį, aišku, galima spręsti įprastais tiesinio programavimo metodais, bet geriau taikyti efektyvesnius algoritmus. Vieną iš tų algoritmų pademonstruosime, spręsdami šitokio varianto trumpiausio kelio uždavinį.

| iš | į |   |   |
|----|---|---|---|
|    | 1 | 2 | m |
| 0  | 5 | 4 | — |
| 1  | — | 3 | 6 |
| 2  | 2 | — | 8 |

Sakykime, tinklą sudaro keturi miestai, sujungti keliais, kurių ilgiai nurodyti paveiksle ir lentelėje.



Toks uždavinys skaičiuojamas dviem etapais pagal paprastą kombinacinę schemą.

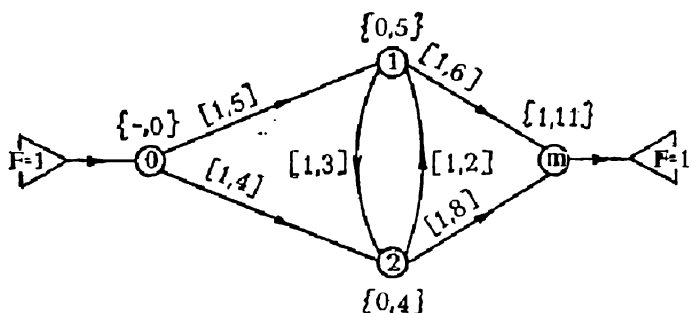
### Trumpiausio kelio pasirinkimo algoritmas

**1 etapas.** Kiekvienam tarpiniam punktui  $i$  priskirsime kortelę, pavyzdžiui,  $\{-, d_i\}$ ; čia pirmoji komponentė rodo paskutinį tarpinį punktą trumpiausiam kelyje iš 0 į punktą  $i$ , o antroji — to trumpiausio kelio ilgį. Iš pradžios punktui 0 priskirsime kortelę, kurios  $d_0=0$ ; pirmoji komponentė, suprantama, visą laiką bus —. Visiems kitiems punktam priskirsime korteles, kurių  $d_i=\infty$  (bet kuris labai didelis skaičius) ir pirmoji komponentė —.

**2 etapas.** Pasirinksime bet kurį kelią  $(i, j)$ , kurio  $d_i + c_{ij} < d_j$ , mazgo  $j$  kortelę pakeisime kortele  $\{i, d_i + c_{ij}\}$  ir tą procesą tęsime tol, kol nebeliks nė vieno tokio kelio. Pirmą kartą pakeitę kortelę, nurodysime trumpesnį kelią

iš 0 į punktą  $j$ , einantį per  $i$ ; paskutinį kartą pakeitę, procesą baigsime, o mazgo  $j$  kortelė rodys trumpiausią atstumą nuo punkto 0 iki punkto  $j$ .

Taip spręsdami uždavinį, sužinome, kad trumpiausias kelias iš 0 į  $m$ , kurio ilgis lygus 11, eina per punktus 0, 1,  $m$ . Atitinkamos kortelės pavaizduotos paveiksle.



Transporto uždavinį, apie kurį buvo pasakojama pirmajame skyriuje, lengvai galima interpretuoti kaip pigiausio srauto tinklų uždavinį. Paveiksle, pateiktame 21 puslapyje, skaitytojas mintyse turėtų taip išdėstyti siuntimo ir paskirties punktus, kad būtų beveik išpildyti pigiausio srauto uždavinio reikalavimai. Siuntimo punkte yra  $F=25$  šaldytuvai, iš kurių 11 reikia nugabenti į pirmąją fabriką, o 14 — į antrąją. Paskirties punktas turi gauti 25 šaldytuvus, būtent, 10 iš pirmosios parduotuvės, 8 iš antrosios ir 7 iš trečiosios. Tos sąlygos sudarys matematinio modelio dalį, kai jį papildysime reikalavimais iš siuntimo punkto į atitinkamus fabrikus nugabenti 11 ir 14 šaldytuvų, o į paskirties punktą — 10, 8 ir 7 šaldytuvus iš atitinkamų parduotuvių. Tokio tipo sąlygos truputį pakeis bendrąjį algoritmą, ir jas galima bus prijungti prie bendrojo pigiausio srauto.

### Komivojažieriaus uždavinys

Kovinis komivojažieriaus šūkis „Geriau pažink savo rajoną!“ mums primena, kad komivojažierius turi gerai žinoti, kur ieškoti klientų, taip pat kokioms prekėms ir kokiam stiliui teikia pirmenybę nuolatiniai ir būsimieji pirkėjai. Štai jau daug metų Naivuolių Baldų Kompanijos komivojažierius Vilis Saimonas keliauja po plačią terito-

riją, stengdamasis pažinti šalį ir pasiekdamas įspūdingų prekių pardavimo rekordų. Bet ne mažiau įspūdingos ir jo išlaidos. Konsultantai valdymo klausimais nusprendė, pasitaikius progai, pasidomėti ir Vilio reikalais. Tačiau Vilį bematant nuvargino šitokio pobūdžio klausimai: „Vili, o kas atsitiko birželio trečiąją Bostone? Tą dieną išlaidos buvo pernelyg didelės“. Vienas iš tų vaikinių „prilipo“ prie Vilio. Keliaujant vaikinys kopijavo visus dokumentus ir užrašus. Norėdamas, kad konsultanto ataskaita būtų palankesnė, Vilis pamėgino susidraugauti su juo. Sužinojęs, kad konsultantas yra matematikas, Vilis pasirausė savo komivojažieriškame istorijų bei jumoreskų bagaže ir „pavaišino“ jį šitokiu anekdotu.

### Arklio protas

Vienas matematikos profesorius nutarė atostogas praleisti kaime. Vietinis fermeris, sužinojęs, kad jo naujasis kaimynas yra matematikas, atėjo pas jį ir papasakojo apie savo arklį, mokantį keturis aritmetikos veiksmus. Profesorius, norėdamas pasijuokti iš fermerio, nuėjo pasižiūrėti to arklio. Fermeris nemelavo — arklys iš tikrųjų mokėjo



skaičiuoti. Jis teisingai sprendė paprastus uždavinius: arba reikalingą skaičių kartų pabelsdavo kanopa į arklidės grindis, arba, sukandęs kreidą dantimis, atsakymą parašydavo lentoje. Profesorius nusprendė tą arklių mokyti ir greitai išmokė jį algebros pradmenų ir Euklido geometrijos.

Pasibaigus atostogoms, profesorius pasiėmė arklių į universitetą, ir šis tapo studentu. Per pirmąjį semestrą arklys lengvai išstudijavo trigonometriją. Ir antrąjį semestrą jis gaudavo vien penketus, bet tik tol, kol nesusidūrė su analizine geometrija. Čia vargšas arklys ir suklupo. Jis niekaip neįstengė suprasti to dalyko. Nieko nepajėgė padaryti nė specialiai priskirti repetitoriai.

Matematikos fakultetas, bijodamas netekti savo geriausio studento, sušaukė skubų susirinkimą, kad nutartų, ką daryti. Buvo pakviesti konsultantai — veterinarai, psichiatrai, žokėjai, bet ir tie neįstengė nieko patarti. Dar kartą peržiūrėjo arklio studijų knygutę: iš aritmetikos, algebros, Euklido geometrijos, trigonometrijos — „labai gerai“, bet analizinė geometrija arkliui, matyt, buvo nepramušama siena.

Pagaliau, ilgai galvojęs, vienas asistentas nuostabiai išsprendė tą dilemą. „Mums jau seniai reikėjo suprasti, — tarė jis, — kad, mėgindami išmokyti arklių analizinės geometrijos, Dekartą statome pirma arklio\*“.

Vilio pasakojimas konsultantui nepadarė nė mažiausio įspūdžio. Vilis nesitvėrė pykčiu ir per pirmąjį apsilankymą firmos kontoroje susikaupusį pyktį išliejo broliui, prezidentui Saimonui. Jis pagrasino mesias darbą, jeigu jo metodų ir išlaidų nepaliks ramybėje. Tačiau prezidentas Saimonas nuramino Vilį, įtikinęs, kad konsultantai valdymo klausimais pritarė visiems jo veiksmams, išskyrus vieną smulkmeną. Iš tyrimo rezultatų matyti, kad Vilis per daug sugaišta laiko ir išleidžia pinigų, važinėdamas iš vieno miesto į kitą. Matyt, jis neplanuoja kelionės ir nesirūpina, kad nuvažiuotas atstumas būtų kuo trumpesnis.

---

\* Žodžių žaismas: angliškai *to put the cart befor the horse* — „kinkyti vežimą prieš arklių“, t. y. ką nors daryti visiškai neteisingai. Angliškąjį *the cart* („vežimas“) amerikiečiai taria beveik taip pat, kaip ir žodį „Decart“. Dekartas (1556—1650) — įžymus prancūzų filosofas ir matematikas, analizinės geometrijos pradininkas. — *Rus. vert. past.*

— Vienintelis dalykas, ko tavęs, Vili, prašome,— tarė prezidentas Saimonas,— tai lankyti miestus tokia eile, kokią nurodys konsultantai. Kitos išlaidos — tavo nuožiūra.

Pajutęs palengvėjimą, Vilis sutiko veikti pagal planą, kai tik jį gaus (iki šiol jis tebelaukia to plano). Panašu, jog paskutinis juoksis Vilis, nes paaiškėjo štai kas: nors ir galima komivojažieriaus uždavinį suformuluoti kaip tiesinio programavimo, bet Vilio teritorija yra tokia didelė (joje yra 1000 miestų, gyvenviečių ir kaimų), kad tas uždavinys pasidaro pernelyg sunkus ir negalima visko apskaičiuoti.

Patikslindami tai, kas pasakyta, komivojažieriaus uždavinį suformuluosime šitaip.

Komivojažierius turi aplankyti visus tam tikros teritorijos miestus. Jis išvyksta iš to miesto, kuriame gyvena, aplanko kiekvieną miestą po vieną kartą ir grįžta namo. Reikia sudaryti maršrutą, kuriuo keliaujant bendrasis komivojažieriaus nuvažiuotas kelias būtų trumpiausias.

Tą uždavinį galima spręsti daugeliu būdų kaip tiesinio programavimo uždavinį. Tačiau matematiniu požiūriu visi tie būdai yra per daug sudėtingi, todėl čia jų neaprašinėsime. Taip formuluojant uždavinį, kintamieji įgyja tik sveikąsias reikšmes 0 ir 1, o lygčių skaičius būna labai didelis. Pavyzdžiui, sprendžiant Vilio uždavinį, reikėtų 10 000 lygčių.

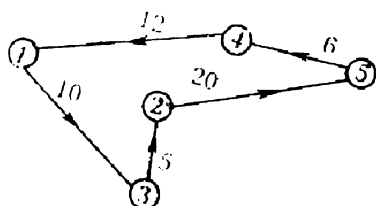
Kokie sunkumai iškyla, sprendžiant tokius uždavinius, matysime iš šio pavyzdžio. Komivojažierius turi aplankyti penkis miestus, kuriuos žymėsime skaitmenimis 1, 2, 3, 4, 5; skaitmeniu 1 pažymėtas miestas, kuriame gyvena komivojažierius. Sakykime, atstumai tarp visų miestų yra žinomi ir iš kiekvieno miesto galima nuvažiuoti į bet kurį kitą miestą. Kad mūsų samprotavimai būtų bendresni, tarsime, jog atstumas, pavyzdžiui, nuo miesto 2 iki miesto 5 nėra lygus tam atstumui, kurį reikia nuvažiuoti, vykstant priešinga kryptimi — iš miesto 5 į miestą 2. Nesimetriška situacija gali susidaryti, pavyzdžiui, dėl vienos krypties judėjimo, apvažiavimų ir pan.

Atstumai tarp tų penkių miestų kilometrais surašyti šioje lentelėje:

| Miestas | 1  | 2  | 3  | 4  | 5  |
|---------|----|----|----|----|----|
| 1       | 0  | 17 | 10 | 15 | 17 |
| 2       | 18 | 0  | 6  | 12 | 20 |
| 3       | 12 | 5  | 0  | 14 | 19 |
| 4       | 12 | 11 | 15 | 0  | 7  |
| 5       | 16 | 21 | 18 | 6  | 0  |

Pavyzdžiui, atstumas nuo miesto 1 iki miesto 5 yra  $d_{15}=17$ , o atstumas nuo miesto 5 iki miesto 1 —  $d_{51}=16$ .

Komivojažierius iš miesto 1 gali vykti į bet kurį iš kitų keturių miestų; sakykime, jis važiuoja į miestą 3.



Iš miesto 3 jis gali važiuoti į bet kurį iš trijų likusių miestų (2, 4 ir 5), pavyzdžiui, į miestą 2. Paskui jis važiuoja į miestą 5, po to, aišku, į miestą 4 ir grįžta į miestą 1.

Tokią miestų eilę (1, 3, 2, 5, 4, 1) vadinsime „ciklu“. Kiekvieną ciklą atitinka tam tikras

kelių ilgis. Minėtąjį ciklą atitinka bendrasis kelių ilgis, lygus

$$d_{13} + d_{32} + d_{25} + d_{54} + d_{41} = 10 + 5 + 20 + 6 + 12 = 53 \text{ (km)}.$$

Penkių miestų uždavinyje galimi  $4 \times 3 \times 2 \times 1 = 24$  ciklai. Spręsdami tokius nesudėtingus uždavinius, galime ištirti visus ciklus bei juos atitinkančius atstumus ir nurodyti tą ciklą, kurį atitinka trumpiausias atstumas. Tačiau, net ir nežymiai padidinus miestų skaičių, toks metodas pasidaro nepraktiškas. Kai reikia aplankyti 10 miestų, yra 362 800 galimų ciklų, o 15 miestų — net 87 178 291 200 ciklų.

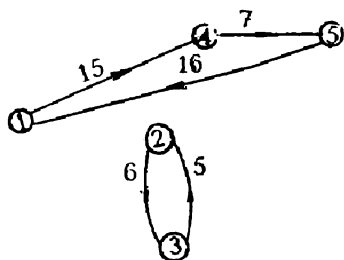
Tas uždavinys tiesinio programavimo metodais sprendžiamas panašiai, kaip ir personalo skirstymo uždavinys. Pasirinkdami ciklą, kiekvienam miestui priskiriame po vieną kelią, jungiantį tą miestą su kitu miestu; be to, iš kiekvieno miesto galima išvykti tik vienu keliu (žmogų leidžiama skirti tik į vieną darbą, be to, jis būtinai turi būti kur nors paskirtas). Kelių, išeinančius iš pirmojo miesto, žymėsime  $x_{12}$ ,  $x_{13}$ ,  $x_{14}$ ,  $x_{15}$  ir turėsime lygtį

$$x_{12} + x_{13} + x_{14} + x_{15} = 1.$$

Nulinė kintamojo reikšmė reiškia, kad tuo keliu nesinaudojama; reikšmė, lygi vienetui,— kad tuo keliu naudojasi. Anksčiau minėtam ciklui sudarysime šitokią lentelę:

| Miestas | 1 | 2 | 3 | 4 | 5 |
|---------|---|---|---|---|---|
| 1       | 0 | 0 | 1 | 0 | 0 |
| 2       | 0 | 0 | 0 | 0 | 1 |
| 3       | 0 | 1 | 0 | 0 | 0 |
| 4       | 1 | 0 | 0 | 0 | 0 |
| 5       | 0 | 0 | 0 | 1 | 0 |

Sunkumas, išskylantis, komivojažieriaus uždavinį traktuojant kaip personalo skirstymo, yra tas, kad optimalus sprendinys, kurį atitinka minimalus kelio ilgis, nebūtinai yra ciklas. Bendrasis kelio ilgis, atitinkantis čia pateikiamą lentelę, yra lygus 49 kilometrams, t. y. mažesnis už 53 kilometrus, atitinkančius anksčiau minėtą ciklą, bet, kaip pavaizduota paveiksle, tokį sprendinį sudaro du „daliniai ciklai“.



Galime prirašyti lygčių, kurios pašalins tokią galimybę. Štai kodėl, sprendžiant tą uždavinį tiesinio programavimo metodais, reikia tiek daug lygčių.

| Miestas | 1 | 2 | 3 | 4 | 5 |
|---------|---|---|---|---|---|
| 1       | 0 | 0 | 0 | 1 | 0 |
| 2       | 0 | 0 | 1 | 0 | 0 |
| 3       | 0 | 1 | 0 | 0 | 0 |
| 4       | 0 | 0 | 0 | 0 | 1 |
| 5       | 1 | 0 | 0 | 0 | 0 |

Tačiau Vilis Saimonas neilgai džiaugsis. Komivojažieriaus uždavinį galima spręsti ir kitais metodais, nors pačios didžiausios apimties išspręstame uždavinyje figūravo tik 70 miestų. Dinaminio programavimo metodu, matyt, galima spręsti tik nedidelės apimties uždavinius, ku-

riuose miestų skaičius ne didesnis kaip 13. Uždavinys, apimantis 70 miestų, buvo sprendžiamas, naudojantis specialia schema, o tiesinio programavimo metodais buvo sprendžiamas uždavinys, apimantis 42 miestus. Vilio uždavinį galima suskirstyti į smulkesnius, arti esančius miestus laikant vienu miestu. Tačiau, kaip ir visais tokiais atvejais, analitikai, apsiginklavę algoritmais ir ESM, turi įrodyti, kad matematiškai surasti sprendiniai yra geresni ir kad jais galima naudotis praktikoje.

Skaitytojui siūlome ištirti, ar nurodytąjį ciklą (1, 3, 2, 5, 4, 1) atitinkantis kelias yra minimalus, o jeigu ne, tai kokį ciklą atitiks minimalus kelias. Komivojažieriaus uždavinio sąvokas galima pritaikyti ir kitoms sritims (pašto pervežimui, optimalios mokinių pervežimo schemos sudarymui). Nereikia bijoti tokių uždavinių dėl pasitaikančių skaičiavimo sunkumų. Čia vis dėlto galima šį tą pasiekti, o kartais ir įdomią istoriją išgirsti.

## Kontraktų uždavinys

*„Parduota!“*

*Aukcionieriaus šūksnis*

Kad ir koks optimumo uždavinys pasitaikytų, aišku, norime pasirinkti tokį leistiną sprendinį, kuris kartu būtų ir optimalus. Siekiame viską daryti kuo geriau (tik ar visada iš tikrųjų taip darome?). Taikydami tiesinio programavimo skaičiavimo metodus, pasirenkame kurį nors sprendinį ir ieškome geresnio už pasirinktąjį, paskui — dar geresnio ir t. t. Todėl savaime kyla klausimas: ar verta naudoti tą visą techniką vien tam, kad sutaupyti viena kitą nelaimingą pensą? Juk išlaidos, ieškant geresnio sprendinio, gali būti gerokai didesnės, negu bus sutaupyta, panaudojus tą sprendinį. Ar verta ieškoti pigiausios dietos? Gal visiškai pakanka tiesiog geros dietos, tenkinančios mūsų poreikius maistingųjų medžiagų atžvilgiu? Ar gamybai planuoti neužtenka Sido Saimono sprendinių, ar iš tikrųjų reikia taupyti?

Kai kuriems uždaviniams iš tiesų pakanka tik gero sprendinio, nors, apskritai kalbant, ir tokiais atvejais, jei užtenka lėšų, reikėtų surasti ir panaudoti optimalų. Tačiau yra uždavinių, kur optimalų sprendinį rasti būtina.

Pavyzdžiui, JAV paruošų žinybai Kongresas įsakęs, sudarinėjant kontraktus su privačiomis firmomis, minimizuoti išlaidas, kitaip sakant, siekti, kad bendros sąnaudos būtų mažiausios.

Kontraktų sudarymas gali būti labai painus valdymo uždavinys, o apribojimai, kuriuos kelia tiekėjai,— labai



sudėtingi. Kadangi, sudarius kontraktus, visas firmų pasiūlas reikia publikuoti, tai asmenys, atliekantys atitinkamus skaičiavimus, turi būti labai įžvalgūs, manipuliuodami tais duomenimis. Tą sunkią naštą padeda nusimesti tiesinis programavimas. Kaip tokie sunkumai įveikiami, naudojantis tiesiniu programavimu? Minėtą uždavinį išnagrinėsime smulkiau.

Sakykime, firma siūlo kokios nors rūšies paruošas, nurodydama:

- 1) prekių arba prekės vieneto kainą;
- 2) maksimalų arba minimalų kiekvieno pavadinimo ir nurodytos kainos vienetų skaičių, kurį firma gali pagaminti;
- 3) įvairias kitas sąlygas, kurias ji mato reikalą iškelti.

Pasiūlos atspindi firmos siekimą gauti pelną, jos įžvalgumą konkurentų pasiūlų atžvilgiu ir jos savus specifinius apribojimus.

Nagrinėdama tas pasiūlas, paruošų žinyba prie kainų, nurodytų kiekvieno tiekėjo, turi pridėti įvairias papildomas išlaidas, pavyzdžiui, transporto išlaidas. Analogiškai iš tų kainų atimamos visos sąnaudos, kurių išvengiama tam tikromis specialiomis sąlygomis. Pavyzdžiui, išlaidos gali sumažėti, perkant išsimokėtinai. Kontraktų uždavinys iš esmės formuluojamas kaip transporto uždavinys, nors dėl kai kurių sąlygų, kurias kelia tiekėjai, kartais tenka jį gerokai pakeisti. Pavyzdžiui, firma galėjo pareikalauti, kad žinyba užsakytų ne mažiau kaip 500 vienetų ir ne daugiau kaip 1000 vienetų, arba galėjo nurodyti skirtingas kainas kiekvienai 1000 vienetų partijai, atsižvelgdama į parduotų prekių kiekį. Sekantis pavyzdys yra paruošų žinybai praktikoje pasitaikęs kontraktų uždavinys. Pateikiame tik to uždavinio duomenis, o skaitytojui siūlome parašyti lygtis ir mėginti surasti sprendinį.

Žemiau pateiktoje lentelėje išvardytiems sandėliams reikia tam tikro skaičiaus nurodytos markiruotės įpakavimo dėžių.

| Sandėlio vieta | Reikalaujama     |           |
|----------------|------------------|-----------|
|                | vidaus reikalams | eksportui |
| Kolumbas       | 10 395           |           |
| Ričmondas      | 12 420           |           |
| San Antonijas  | 10 395           |           |
| Skenektadis    | 9 720            | 39 555    |
| Juta           | 3 240            |           |
| Šarpas         | 5 535            |           |
| Oburnas        | 3 645            |           |
| Atlanta        | 10 330           | 3 510     |
| Iš viso        | 65 680           | 43 065    |
| Bendra suma    | 108 745          |           |

Dėžės skirstomos į dvi kategorijas: vidaus reikalams ir eksportui, nes specialus įpakavimas eksportui yra brangesnis.

Keturi tiekėjai pateikia pasiūlymus. Tiekėjas 1 reikalauja užsakyti ne mažiau kaip 11 500 dėžių, be to, ne daugiau kaip 33 145 dėžės vidaus reikalams ir ne daugiau kaip 3510 dėžių eksportui. Tiekėjas 2 neriboja maksimalaus ir minimalaus dėžių skaičiaus. Tiekėjas 3 siūlo ne daugiau kaip 60 000 dėžių, o tiekėjas 4 — tik dėžės vidaus reikalams ir nenurodo jokių kitų sąlygų. Visa ši informacija, taip pat vienos dėžės kaina kiekvieno atveju pateikta lentelėje:

| Miestas                        | Reikalaujama | Maksimalus skaičius pasiūlytas |           |           |           |           |
|--------------------------------|--------------|--------------------------------|-----------|-----------|-----------|-----------|
|                                |              | tiekėjo 1                      |           | tiekėjo 2 | tiekėjo 3 | tiekėjo 4 |
|                                |              | vidaus reikalams               | eksportui |           |           |           |
|                                |              | 33 145                         | 3 510     | 108 745   | 60 000    | 65 680    |
| Kolumbas                       | 10 395       | 0,7289                         | —         | 0,6868    | 0,6574    | 0,6832    |
| Ričmondas                      | 12 420       | 0,7398                         | —         | 0,7058    | 0,6489    | 0,6724    |
| San Antonijas                  | 10 395       | 0,7229                         | —         | 0,7204    | 0,6904    | 0,7227    |
| Skenektadis (vidaus reikalams) | 9 720        | 0,7406                         | —         | 0,7075    | 0,6318    | 0,6627    |
| Skenektadis (eksportui)        | 39 555       | —                              | 0,7749    | 0,7319    | 0,6452    | —         |
| Juta                           | 3 240        | 0,7247                         | —         | 0,7358    | 0,6944    | 0,7306    |
| Sarpas                         | 5 535        | 0,7276                         | —         | 0,7389    | 0,6973    | 0,7339    |
| Oburnas                        | 3 645        | 0,7297                         | —         | 0,7389    | 0,6973    | 0,7339    |
| Atlanta (vidaus reikalams)     | 10 330       | 0,7325                         | —         | 0,7049    | 0,6646    | 0,6917    |
| Atlanta (eksportui)            | 3 510        | —                              | 0,7663    | 0,7291    | 0,6816    | —         |

Nurodytos kainos apima ir pristatymo išlaidas. Pavyzdžiui, norint nupirkti vieną dėžę iš tiekėjo 2 ir nugabenti ją į Kolumbo sandėlį, reikia 0,6868 dolerio. Sprendinys, tenkinantis visas tiekėjų iškeltas sąlygas ir minimizuojantis bendrąsias paruošų žinybos išlaidas, pateiktas šioje lentelėje:

| Miestas       | Užsakomų dėžių skaičius |            |            |            |
|---------------|-------------------------|------------|------------|------------|
|               | tiekėjui 1              | tiekėjui 2 | tiekėjui 3 | tiekėjui 4 |
| Kolumbas      | —                       | —          | —          | 10 395     |
| Ričmondas     | —                       | —          | —          | 12 420     |
| San Antonijas | —                       | 10 395     | —          | —          |

| Miestas          | Užsakomų dėžių skaičius |            |            |            |
|------------------|-------------------------|------------|------------|------------|
|                  | tiekėjui 1              | tiekėjui 2 | tiekėjui 3 | tiekėjui 4 |
| Skenektadis      |                         |            |            |            |
| vidaus reikalams | —                       | —          | 4 515      | 5 205      |
| eksportui        | —                       | —          | 39 555     | —          |
| Juta             | —                       | —          | 3 240      | —          |
| Sarpas           | —                       | —          | 5 535      | —          |
| Oburnas          | —                       | —          | 3 645      | —          |
| Atlanta          |                         |            |            |            |
| vidaus reikalams | —                       | —          | —          | 10 330     |
| eksportui        | —                       | —          | 3 510      | —          |

Bendra išlaidų suma šiuo atveju yra lygi 72 953,1935 dolerio. Tiesinio programavimo metodai, kuriais buvo rastas šis sprendinys, laiduoja, kad geresnio rezultato pasiekti neįmanoma.

## Lošimų teorija

*Lošimas — dalykas rimtas*

Galima sakyti, kad lošimų teorija, kaip ir tiesinis programavimas, yra dabartinės matematikos sritis. Nespecialistui gali pasirodyti, kad tuo jų panašumas ir baigiasi, nes tiesinio programavimo metodais ieškome, kaip efektyviausiai paskirstyti ribotus išteklius, o lošimų teorijoje — tokios strategijos, kurios laikydamiesi galime daugiausiai išlošti. Vis dėlto tarp tų uždavinių yra nuostabus sąryšis, pasireiškiantis tuo, kad tam tikros lošimų teorijos uždavinių klasės matematinis modelis sutampa su tiesinio programavimo modeliu.

Bendriausias uždavinys, sprendžiamas lošimų teorijoje, yra šitoks:  $n$  lošėjų (žymėsime juos  $P_1, P_2, \dots, P_n$ ) lošia tam tikrą lošimą. Kaip turi lošti kiekvienas iš jų, kad pasiektų geriausią rezultatą? Pasibaigus partijai, kiekvienas lošėjas gauna tam tikrą pinigų sumą, vadinamąją „išlošį“. Pralošusio asmens tas „išlošis“ lygus monetų srautui, „ištekėjusiam“ iš jo kišenės į laimingesniųjų partnerių kišenes. Jeigu visi pinigai, ir išloštieji, ir praloštieji, lieka pas lošėjus, tai sakoma, kad tai — „nulinės sumos“ lošimas. Nenulinės sumos lošimo pavyzdžiu gali būti po-

keris, kai lošimo namai ima tam tikrą procentą nuo kiekvieno statymo.

Lošimai, be to, klasifikuojami pagal lošėjų skaičių ir pagal galimų ėjimų skaičių. Pavyzdžiui, šachmatai yra dviejų asmenų lošimas su baigtiniu ėjimų skaičiumi (laikantis kokios nors „sustabdymo taisyklės“), o pokerį gali lošti daug lošėjų, bet ėjimų skaičius taip pat bus baigtinis. Dvikova, kai abu dalyviai gali šauti bet kuriuo tam tikro laiko intervalo momentu, yra dviejų asmenų lošimas su begaliniu galimų ėjimų skaičiumi. Be to, lošimas gali būti kooperatinis arba nekooperatinis. Pirmuoju atveju lošėjai gali sudaryti grupes ir veikti kolektyviai, o antruoju kiekvienas lošia pats sau. Suprantama, dviejų asmenų lošimas negali būti kooperatinis. Čia aptarsime baigtinius dviejų asmenų nulinės sumos lošimus, nes tik tokie lošimai yra susiję su tiesiniu programavimu.

Kaip pirmąjį pavyzdį aprašysime praktikoje pasitaikiusios strateginės situacijos analizę, kuri jau dabar yra tapusi klasikinė\*.

### **Dviejų strategijų lošimas**

1943 metų vasario—kovo mėnesiais Japonijos laivų konvojus susirinko Rabaule (Naujosios Britanijos sala), ruošdamasis plaukti į Laję (Naujosios Gvinėjos sala). Amerikiečių karinė vadovybė nutarė pulti tą konvojų lėktuvais ir padaryti jam kuo daugiau žalos. Japonijos konvojaus vadas turėjo dvi galimybes: jo pasiūstas 16 laivų konvojus Naująją Britaniją galėjo apiplaukti arba šiauriniu, arba pietiniu keliu. Abiem atvejais reikėjo plaukti tris dienas. Pasirinkti kelią reiškė pasirinkti vieną iš dviejų galimų strategijų. Pagal meteorologinius duomenis (juos žinojo abi šalys), šiauriniame kelyje laukiamas lietus ir blogas matomumas, o pietiniame — giedras oras.

Amerikiečių vadas irgi turėjo dvi galimybes. Didesnę žvalgybinių lėktuvų dalį jis galėjo sutelkti arba pirmame, arba antrame kelyje. Aptikus konvojų, bombonešiai tuoj pat galėjo smogti jam ir viename, ir kitame kelyje.

Amerikiečių vado tikslas — kuo greičiau aptikti konvojų ir kuo daugiau dienų jį bombarduoti; japonų vadas

---

\* O. G. Haywood. Jr. Military Decision and Game Theory.— „J. Oper. Res. Soc. Am.“, 1954, vol. 2, No 4.

siekė bombardavimų skaičių sumažinti iki minimumo. Abu savo problemą sprendė nepriklausomai vienas nuo kito. Kaip turi pasielgti kiekvienas iš jų? Prieš pasirinkdami reikalingą strategiją, pateiksime šiek tiek papildomų duomenų.

Turėdamas duomenis apie oro būklę, lėktuvų mobilumą ir kai kuriuos kitus apribojimus, amerikiečių vadas galėjo sudaryti lentelę ( $2 \times 2$ ), arba, kaip mes sakysime, „išlošių matricą“, kurioje nurodyta, kiek dienų tęsis bombardavimas, žiūrint kuriuo keliu bus pasiųsti žvalgybiniai lėktuvai. Analizuojama maždaug šitaip.

1. Amerikiečių strategija — žvalgybinius lėktuvus sutelkti šiauriniame kelyje.
  - a. Konvojus pasiųstas šiauriniu keliu.  
Blogas oras trukdo žvalgybai, bet, teisingai paskirsčius žvalgybinius lėktuvus, galima bombarduoti dvi dienas.
  - b. Konvojus pasiųstas pietiniu keliu.  
Giedra, bet dėl riboto žvalgybinių lėktuvų skaičiaus bombarduoti bus galima ne daugiau kaip dvi dienas.
2. Amerikiečių strategija — žvalgybinius lėktuvus sutelkti pietiniame kelyje.
  - a. Konvojus pasiųstas šiauriniu keliu.  
Dėl blogo matomumo ir netikusio žvalgybinių lėktuvų paskirstymo bombarduoti teks tik vieną dieną.
  - b. Konvojus pasiųstas pietiniu keliu.  
Viskas susiklosto labai sėkmingai (geras oras, daug žvalgybinių lėktuvų); konvojų galima bombarduoti tris dienas.

Iš tų duomenų sudarome išlošių matricą.

|                                                               |                  | Japonų strategija<br>(kelio pasirinkimas) |                    |
|---------------------------------------------------------------|------------------|-------------------------------------------|--------------------|
|                                                               |                  | Siaurinis<br>kelias                       | Pietinis<br>kelias |
| Amerikiečių strategija<br>(žvalgybinių lėktuvų<br>sutelkimas) | Siaurinis kelias | 2 dienos                                  | 2 dienos           |
|                                                               | Pietinis kelias  | 1 diena                                   | 3 dienos           |

Norint tokią karinę situaciją nagrinėti kaip lošimų teorijos uždavinį, reikia tarti, kad japonų vadas turi tuos pačius duomenis (gali parašyti tą pačią išlošių matricą ir interpretuoti ją taip pat, kaip ir mes). Tai — nulinės sumos lošimas, nes bombardavimo diena, laimėta amerikiečių, yra pralošta japonų, t. y. baigtis, gera vienam vadui, yra bloga kitam. Sunkumai lošimų teorijoje iš dalies kyla todėl, kad reikia suderinti varžovų požiūrius į išlošių matricą sudarančių skaičių tikrąsias reikšmes. Svarstydami tą klausimą, turėtume gerokai nukrypti į šalį, todėl palikime jį lošimų teorijos traktatams.

Analizuodami šį paprastą dviejų strategijų lošimą, pastebime daug būdingų lošimų teorijos bruožų. Aptariamąją išlošių matricą parašę, kaip įprasta matricų teorijoje, lošimą laikysime matrica  $(2 \times 2)$ , kurios eilutės atitinka maksimizuojančio lošėjo (aptariamuoju atveju — amerikiečių vado) strategijas, o stulpeliai — minimizuojančio lošėjo (japonų vado) strategijas. Išlošių matrica yra šitokia:

$$\begin{pmatrix} 2 & 2 \\ 1 & 3 \end{pmatrix}.$$

Dabar pažvelkime į vadus, kurie sėdi savo štabuose, žiūri į skaitmenis ir stengiasi nuspręsti, ką daryti. Japonų vadas, labai geras go lošėjas, yra įsitikinęs, kad jo pasirinkta strategija bus geriausia. Vis dėlto išlošių matrica jam kelia nerimą: kad ir ką jis darytų, konvojus, matyt, bus aptiktas ir užpultas bombonešių. Situacija kaip tik tokia, kai išlošti neįmanoma. Geriausia, ko jis gali tikėtis, — kad bus bombarduojama tik vieną dieną. Ar jam pavyks tai pasiekti?

Lygindamas stulpelius (išlošį, pasirinkus šiaurinę strategiją, su išlošiu, pasirinkus pietinę), japonų vadas pastebi, kad negalima rinktis pietinio kelio. Jeigu pasirinktų jį, tai konvojus būtų bombarduojamas dvi ar net tris dienas, tuo tarpu pasirinkus šiaurinį kelią, atitinkamų dienų skaičius būtų lygus dviem arba vienam. Japonų vadas pasirenka tą strategiją, remdamasis savo matrica. Jis galėtų tuos duomenis pasiųsti pačiam amerikiečių vadui, nes pastarasis, taip pat strategas, tą patį stulpelį pasirinktų iš savo matricos.

Amerikietis kvadratinę matricą pakeičia matrica, sudaryta iš dviejų eilučių, atitinkančių jo strategijas, ir iš vieno stulpelio, reiškiančio, kad japonai turi pasirinkti šiaurinį kelią. Naujoji išlošių matrica yra šitokia:

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Dabar uždavinys pasidarė visiškai paprastas. Kadangi amerikiečių vadas nori maksimizuoti bombardavimo dienų skaičių, tai jis turi rinktis šiaurinį kelią. Ir tarsi tai būtų matematikų apskaičiuota, jis iš tikrųjų žvalgybinius lėktuvus sutelkė šiauriniame kelyje.

Nagrinėtasis uždavinys turi vieną ypatybę (sakoma, kad uždavinys išspręstas, naudojantis grynąja strategija), ir štai kokią: abu vadai buvo priversti rinktis šiaurinį kelią. Vadui, arba lošėjui, net nėra reikalo nagrinėti kitas strategijas (apsisprendžiant jam nereikia mesti monetos). Taip yra todėl, kad minėto uždavinio matrica turi vadinamąjį „balno tašką“. Matricos balno tašku vadinamas toks jos elementas, kuris yra mažiausias savo eilutėje ir didžiausias savo stulpelyje (dvi bombardavimo dienos šiaurinio kelio strategijų sankryžoje kaip tik ir yra toks skaičius). Atitinkama eilutė ir atitinkamas stulpelis yra abiejų lošėjų optimalios strategijos, o išlošis sutampa su balno taško reikšme. Tokios situacijos priežastys susijusios su tam tikru lošimų teorijos konservatizmu. Šią mintį paaiškinsime smulkiau.

Tarkime, kad koks nors lošimas apibrėžiamas šitokia  $(3 \times 3)$  matrica:

$$\begin{pmatrix} 3 & 5 & 6 \\ 2 & -1 & 3 \\ 0 & 7 & 4 \end{pmatrix}.$$

Primename, kad išlošių matricoje skaičiai yra surašyti maksimizuojančio lošėjo (pavyzdžiui, amerikiečių vado) atžvilgiu; todėl teigiamas skaičius reiškia, kad laimi tas lošėjas, kurio strategijos sudaro eilutes, neigiamas skaičius reiškia, kad jis tą sumą pralaimi, o nulis — kad pini-

gai visiškai nepereina iš rankų į rankas. Lošėjas, kurio strategijos sudaro eilutes (jį vadinsime pirmuoju lošėju), žiūri į skaičius, užrašytus pirmoje eilutėje, ir pastebi, kad, pasirinkus pirmąją strategiją, blogiausiai jam bus tuo atveju, kai ir antrasis lošėjas pasirinks pirmąją strategiją. Tada pirmasis lošėjas gaus tris vienetus vietoj penkių ar šešių. Pirmasis lošėjas taip pat analizuoja kitas dvi eilutes ir pastebi, kad gali išlošti atitinkamai  $-1$  vienetą (t. y. pralošti 1 vienetą) ir 0 vienetų. Tuos minimaliuosius skaičius užrašysime šalia matricos:

$$\begin{pmatrix} 3 & 5 & 6 \\ 2 & -1 & 3 \\ 0 & 7 & 4 \end{pmatrix} \begin{array}{l} \overline{3} \\ -1 \\ 0 \end{array}$$

Tie papildomi skaičiai rodo, kas blogiausia gali atsitikti pirmajam lošėjui kiekvienu atveju. Pirmasis lošėjas gali pasirinkti geriausią iš nurodytų blogiausių galimybių ir visada laikytis pirmosios strategijos. (Blogiausiu atveju jis išloš bent tris vienetus.) Ką gali padaryti antrasis lošėjas, siekdamas kuo mažiau pralošti?

Analogiškai analizuodami antrojo lošėjo strategijas, pastebime, kad, jam pasirinkus pirmąjį stulpelį, blogiausia bus tuo atveju, kai pirmasis lošėjas pasirinks pirmąją eilutę, t. y. kai antrasis lošėjas turės sumokėti tris vienetus. Pasirinkus antrąjį ir trečiąjį stulpelį, „išlošis“ atitinkamai lygus 7 ir 6. Matricą papildę tais skaičiais, gauname

$$\begin{pmatrix} 3 & 5 & 6 \\ 2 & -1 & 3 \\ 0 & 7 & 4 \end{pmatrix} \begin{array}{l} \overline{3} \\ -1 \\ 0 \end{array} \begin{array}{l} \overline{3} \\ 7 \\ 6 \end{array}$$

Geriausią, ką tokiu atveju gali padaryti antrasis lošėjas (neskaitant atsisakymo lošti),—tai visą laiką laikytis pirmąjį stulpelį atitinkančios strategijos ir nuolat pralošinti po 3 vienetus. Vadinasi, tiriamosios matricos elementas 3 yra balno taškas. Taigi, remiantis analize, analogiškai atliktajai, randami lošimų su balno taškais sprendiniai. Matome, kad lošėjas, nukrypęs nuo savo gryniosios

strategijos (eilutės arba stulpelio), atitinkančios balno taško padėtį, rizikuoja daugiau pralošti arba mažiau išlošti.

Tačiau įdomesni yra tie lošimai, kurie neturi balno taškų. Jeigu mėgintume panašiai analizuoti lošimą, išreikštą  $(3 \times 4)$  matrica

$$\begin{pmatrix} 1 & 5 & 0 & 4 \\ 2 & 1 & 3 & 3 \\ 4 & 2 & -1 & 0 \end{pmatrix},$$

tai pamatytume, kad pirmasis lošėjas pasiekia geriausią iš blogiausių galimybių (vadinamąjį „maksiminą“), pasirinkdamas antrąją eilutę, o antrasis lošėjas realizuoja blogiausią iš geriausių galimybių („minimaksą“), pasirinkdamas trečiąjį stulpelį:

$$\begin{pmatrix} 1 & 5 & 0 & 4 \\ 2 & 1 & 3 & 3 \\ 4 & 2 & -1 & 0 \end{pmatrix} \begin{array}{l} 0 \\ \overline{1} \\ -1 \end{array}$$

$$\begin{array}{cccc} 4 & 5 & \overline{3} & 4 \end{array}$$

Šiuo atveju balno taško nėra. Ir vis dėlto pirmasis lošėjas išloš, ne mažiau kaip 1 vienetą, visada laikydamasis savo gryniosios antrosios strategijos, o antrasis lošėjas praloš, ne daugiau kaip 3 vienetus, laikydamasis savo trečiosios strategijos. Lošiant tokios rūšies lošimus, dalyvių išlošis gali būti didesnis už vidutinį, atsitiktiniu būdu pasirinkus vieną iš galimų strategijų. Tokiu atveju pirmasis lošėjas gali laimėti daugiau kaip 1 vienetą, o antrasis — pralošti mažiau kaip 3 vienetus. Kaitaliojant strategijas ir atsitiktinai pasirenkant vieną iš jų kiekvienai partijai, laukiamojo išlošio dydis gali pasidaryti lygus skaičiui, įterptam tarp maksimino reikšmės 1 ir minimakso reikšmės 3; tokiu atveju kiekvienas lošėjas laimi, palyginus su grynujų strategijų išlošiu. Mišriosios strategijos sąvoką pailustruosime visiems žinomo lošimo „pora—nepora“ pavyzdžiu.

## „Pora—nepora“

Šį lošimą lošia dviese. Pirmasis lošėjas stengiasi atspėti, kokia pinigų suma, lyginė ar nelyginė, yra antrojo lošėjo rankoje („pora“ ar „nepora“). Nagrinėsime šitokį to lošimo variantą: jeigu pirmasis lošėjas atspėja, tai iš antrojo lošėjo jis gauna 1 vienetą (+1), o jeigu neatspėja, tai antrajam moka 1 vienetą (−1). Išlošis −1 reiškia, kad pirmasis lošėjas duoda antrajam 1 vienetą. Kad būtų aiškiau, visa tai pavaizduosime matrica:

|                             |          | Antrojo lošėjo pasirinkimas                      |          |
|-----------------------------|----------|--------------------------------------------------|----------|
|                             |          | „pora“                                           | „nepora“ |
| Pirmojo lošėjo pasirinkimas | „pora“   | $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ |          |
|                             | „nepora“ |                                                  |          |

Matome, kad matrica neturi balno taško („maksiminas“ lygus −1, o „minimaksas“ +1).

Dabar nustatysime kiekvieno lošėjo mišriąsias strategijas. Pirmasis lošėjas „poros“ ir „neporos“ pasirinkimą norėtų kaitalioti atsitiktinai, nes, jeigu jis laikytųsi pastovios taktikos (pavyzdžiui, visada rinktųsi „porą“), tai jo priešininkas, taip pat nekvailas lošėjas (viena iš lošimų teorijos prielaidų), nepraleistų progos tuo pasinaudoti. Sakykime,  $x_1$  yra tikimybė, kad pirmasis lošėjas pasirenks „porą“, o  $x_2$  — tikimybė, kad jis pasirenks „neporą“. Analogiškai apibrėšime tikimybes  $y_1$  ir  $y_2$  antrajam lošėjui. Turėtume tipiską atvejį, jei, pavyzdžiui, pirmojo lošėjo pir-

mosios strategijos pasirinkimo tikimybė būtų lygi  $\frac{3}{4}$ , o antrosios  $\frac{1}{4}$ . Šiuo atveju tikimybė interpretuojama kaip vienos ar kitos strategijos pasirinkimo dažnumas. Jei mišriosios strategijos tikimybės yra  $x_1 = \frac{3}{4}$  ir  $x_2 = \frac{1}{4}$ , pirmasis lošėjas, sulošęs labai daug partijų, pastebėtų, kad maž-

daug tris ketvirčius viso laiko jis laikėsi pirmosios strategijos, o vieną ketvirtį — antrosios. Kadangi suma tikimybių, susijusių su kiekvienu lošėju, turi būti lygi vienetui, tai galima užrašyti  $x_1 + x_2 = 1$  ir  $y_1 + y_2 = 1$ ; čia  $x$  ir  $y$  yra neneigiami skaičiai.

Pagal apibrėžimą nulinės sumos lošimo sprendiniu vadiname optimalių mišriųjų strategijų porą (po vieną kiekvieno lošėjo strategiją) ir tokį skaičių  $v$  (lošimo vertę), kad, pirmajam lošėjui laikantis savo optimalios mišriosios strategijos, o antrajam — bet kokios strategijos, pirmasis lošėjas laimi ne mažiau kaip  $v$ ; jeigu antrasis lošėjas laikosi savo optimalios mišriosios strategijos, o pirmasis — bet kokios, tai antrasis pralošia ne daugiau kaip  $v$ .

Sudarysime šio uždavinio matematinį modelį, bet neįteiksime jam per didelės reikšmės. Skaitytojui teks arba susitaikyti su tuo, arba praleisti keletą puslapių.

Pirmasis lošėjas stengiasi taip maksimizuoti lošimo vertę  $v$ , kad būtų tikimybės  $x_1 + x_2 = 1$  ir kad, laikydamasis strategijos  $(x_1, x_2)$  prieš bet kurią grynąją antrojo lošėjo strategiją, jis loštų mažų mažiausiai  $v$ ; kitaip sakant,

$$x_1 - x_2 \geq v,$$

$$-x_1 + x_2 \geq v.$$

Tas nelygybes užrašėme, nežinomas neneigiamas tikimybes padauginę iš atitinkamų išlošių matricos stulpelio elementų; iš pirmojo stulpelio elementų dauginome šitaip:

$$\begin{array}{c} \boxed{x_1} \times \begin{pmatrix} 1 & -1 \end{pmatrix} \\ + \\ \boxed{x_2} \times \begin{pmatrix} -1 & 1 \end{pmatrix} \end{array}$$

Kiekviena tokia sandaugų suma yra pirmojo lošėjo išlošio matematinė viltis, kai antrasis lošėjas pasirenka atitinkamą grynąją strategiją. Pirmasis lošėjas siekia, kad jo išlošio matematinė viltis būtų bent jau ne mažesnė už nežinomą lošimo vertę  $v$ . Skaičiaus  $v$  ženklas gali būti bet koks. Teigiamas  $v$  reiškia, kad pirmasis lošėjas turi daugiau galimybių išlošti, negu antrasis;  $v=0$  reiškia, kad abu lošėjai turi vienodas galimybes; neigiamas  $v$  reiškia, kad antrasis lošėjas turi daugiau galimybių išlošti, negu pirmasis. Pirmasis lošėjas yra suinteresuotas maksimizuoti  $v$

ir rasti neneigiamas  $x_1$  ir  $x_2$  reikšmes, tenkinančias šias sąlygas:

$$x_1 - x_2 \geq v,$$

$$-x_1 + x_2 \geq v,$$

$$x_1 + x_2 = 1.$$

Skaičius  $v$  taip pat yra šio uždavinio kintamasis, ir vienintelė kliūtis suformuluotąjį uždavinį priskirti prie tiesinio programavimo uždavinių yra ta, kad  $v$  gali įgyti ir teigiamas, ir neigiamas reikšmes. Tą nedidelę anomaliją galima pašalinti įvairiais būdais. Pavyzdžiui, prie kiekvieno išlošių matricos elemento galima pridėti didelį teigiamą skaičių  $w$ , kad visi elementai pasidarytų teigiami. Optimālios strategijos liks tos pačios, o naujoji lošimo vertė  $v + w$  bus teigiamas skaičius.

Analogišką tiesinio programavimo uždavinį gali spręsti ir antrasis lošėjas, t. y. minimizuoti  $v$ , kai

$$y_1 - y_2 \leq v,$$

$$-y_1 + y_2 \leq v,$$

$$y_1 + y_2 = 1,$$

$$y_1 \geq 0,$$

$$y_2 \geq 0.$$

Dydis  $v$  bus tas pats, kaip ir anksčiau. Lošiant „porą—neporą“,  $v=0$ , t. y. tas lošimas yra teisingas — abiejų lošėjų galimybės yra vienodos.

Tiesinio programavimo uždaviniai pirmojo ir antrojo lošėjo atžvilgiu yra glaudžiai susiję, kaip pradinis ir dualusis tiesinio programavimo uždaviniai. (Tuos lošimo uždavinius reikia tik truputį patikslinti, kad jie būtų tikrai dualieji uždaviniai.)

Prisiminę geometrinių tiesinio programavimo uždavinių sprendimą, aptartą 3 skyriuje, lengvai galime spręsti tuos lošimų teorijos uždavinius, kuriuose kiekvienas varžovas gali naudotis tik dviem strategijomis, t. y. lošimus  $(2 \times 2)$ . (Būdą, kurį dabar aprašysime, faktiškai galima taikyti lošimams  $(2 \times m)$ , kai vienas lošėjas gali naudotis tik dviem strategijomis, o kitas —  $m$ .) Čia išspręsimė tik pirmojo lošėjo uždavinį, o skaitytojui paliksime jį išspręsti

antrojo lošėjo atžvilgiu. Taigi nagrinėsime šitokį uždavinį.

Reikia maksimizuoti  $v$ , kai

$$\begin{aligned}x_1 - x_2 &\geq v, \\ -x_1 + x_2 &\geq v, \\ x_1 + x_2 &= 1, \\ x_1 &\geq 0, \\ x_2 &\geq 0.\end{aligned}$$

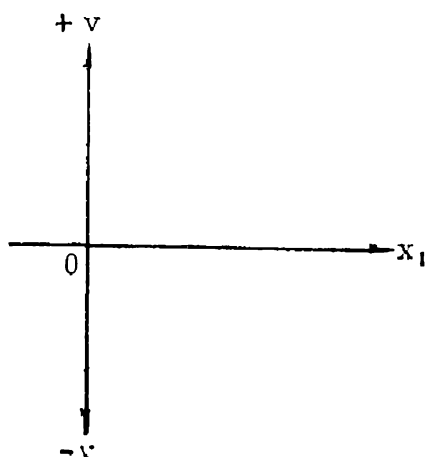
Kadangi geometrinis sprendimo metodas taikomas tik dviejų kintamųjų uždaviniams, tai šį trijų kintamųjų ( $x_1$ ,  $x_2$ ,  $v$ ) uždavinį reikia pakeisti dviejų kintamųjų uždaviniu. Tai lengva padaryti, naudojantis lygtimi  $x_1 + x_2 = 1$ , iš kurios vieną kintamąjį galima išreikšti kitu kintamuoju. Kadangi  $x_2 = 1 - x_1$ , tai nelygybėse vietoj  $x_2$  įrašę jo išraišką kintamuoju  $x_1$ , gauname naujus apribojimus:

$$\begin{aligned}x_1 - (1 - x_1) &\geq v, \\ -x_1 + (1 - x_1) &\geq v, \\ 0 \leq x_1 \leq 1,\end{aligned}$$

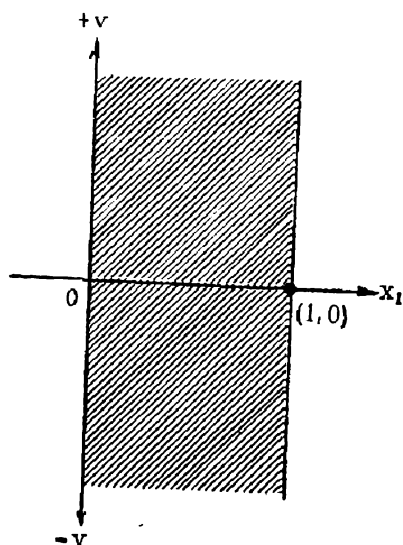
arba

$$\begin{aligned}2x_1 - v &\geq 1, \\ -2x_1 - v &\geq -1, \\ 0 \leq x_1 \leq 1\end{aligned}$$

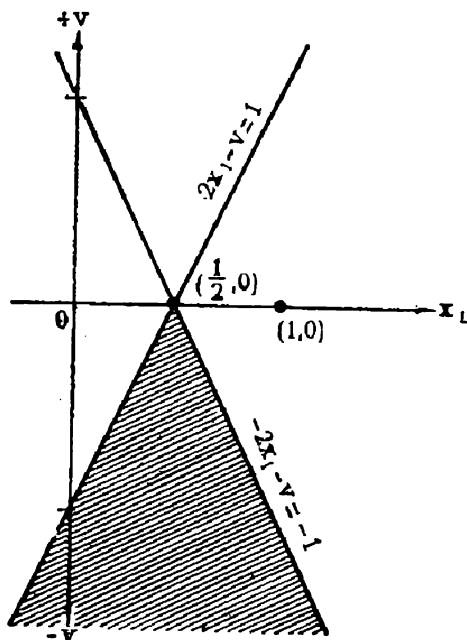
Kadangi tai yra pirmojo lošėjo lošimo modelis, tai stengsimės rasti maksimalią kintamojo  $v$  reikšmę, turėdami omenyje, kad  $v$  (lošimo vertės) ženklas gali būti bet koks.  $x_1$  laikykime viena koordinačių ašimi, o  $v$  — kita.



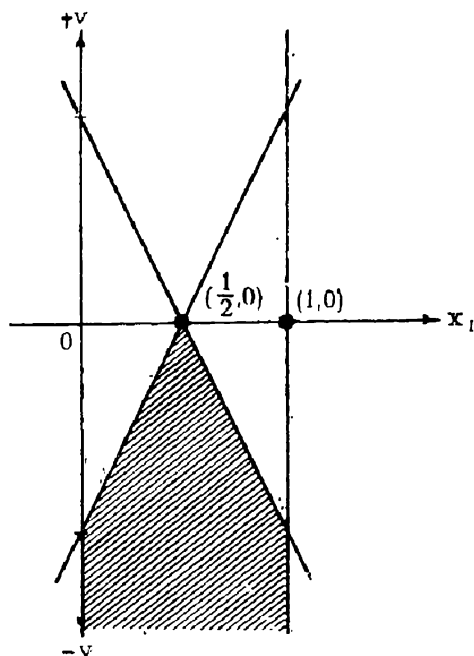
Iš nelygybės  $0 \leq x_1 \leq 1$  išplaukia, kad kintamojo  $x_1$  reikšmė turi būti neneigiamas skaičius, ne didesnis kaip vienetą; paveiksle tai pavaizduota neapibrėžta užbrūkšniuota sritimi.



Dabar nubrėšime tieses  $2x_1 - v = 1$  ir  $-2x_1 - v = -1$  ir gausime abiejų nelygybių sprendinių erdvę. Ta erdvė pavaizduota užbrūkšniuota sritimi šiame paveiksle:



Tuos du brėžinius uždėję vieną ant kito, randame taškus, tenkinančius visus tris sprendžiamojo uždavinio apribojimus.



Šioje užbrūkšniuotoje srityje ieškome taško, kurio koordinatė  $v$  būtų maksimali. Taigi optimalus sprendinys bus taškas  $x_1 = \frac{1}{2}$ ,  $v=0$ . Tas sprendinys yra vienintelis, nes visų kitų sprendinių  $v$  reikšmė yra neigiama. Kadangi  $x_2 = 1 - x_1$ , tai pirmojo lošėjo optimali mišrioji strategija yra šitokia: vieną iš dviejų strategijų jis turi rinktis atsitiktinai; kiekvienos strategijos pasirinkimo dažnumas turi būti lygus  $\frac{1}{2}$ , nes  $x_1 = \frac{1}{2}$ ,  $x_2 = \frac{1}{2}$ . Tokia strategija jam laiduoja vidutinį išlošį, lygų nuliui, todėl lošimas yra teisingas. Antrojo lošėjo optimali strategija yra tokia pat.

#### Lošimas „klėsių tuštinimas“

Ieškodami paskutinio pavyzdžio, vyksime į kasmetinį Naivuolių Kompanijos pikniką (tuo pat metu vyksta ir

kaimo mugė). Tarp svečių, aišku, sukinėjasi ir grupė landdžiųjų konsultantų valdymo klausimais. Vienas iš brolių, Si Saimonas, išmėginęs savo gabumus lošti bingo, ruletę ir kitus azartinius lošimus, ką tik priėjo prie jų. Tiesą sakant, jis sugrįžo namo tik pasiskolinti pinigų. Grįždamas, jau tuščiomis kišenėmis, jis aptiko naują lošimą, kur, kaip jam atrodė, galima labai lengvai išlošti. Tai



buvo lošimas kortomis, ir Si, susipažinęs su jo taisyklėmis, pajuto, kad rado optimalią strategiją. Tačiau nė vienas iš jo brolių, žinojusių, kad Si yra tiesiog apsigimęs nevykėlis (jis buvo viceprezidentas išbrokuotos produkcijos grąžinimo klausimais), nedavė jam pinigų.

Netekęs vilties, jis priėjo prie konsultantų grupės ir papaiškino, ko jam reikia.

— Papasakokite mums apie tą naująjį lošimą,— pareikalavo jie.— Jeigu mūsų analizė patvirtins jūsų žodžius, tai mes jau kaip nors „susimesime“.

— Aš visiškai įsitikinęs, kad sugebėsiu išlošti,— atsakė Si,— o taisyklės yra šitokios. Aš lošiu su „šaukliu“. Abu turime po tris kortas. Jis — būgnų bei kryžių tūzus ir būgnų dviakę. Aš taip pat turiu būgnų ir kryžių tūzus, bet trečioji mano korta — kryžių dviakę. Abu pasirenkame po vieną kortą ir kartu parodome vienas kitam. Jeigu tos kortos yra skirtingų spalvų, tai išlošiu aš, o jeigu vieno — jis. Jeigu abi pasirinktosios kortos yra dviakės, tai niekas neišlošia. Priešingu atveju išlošis lygus akių skaičiui tos kortos, kurią parodė išlošęs\*. Štai ir viskas. Labai paprasta, tiesa? Man reikia sužinoti tik vieną dalyką — kokia eile rinktis kryžius, na, gal būtų, reikėtų keletą kartų parodyti būgnų tūzą. Aš įsitikinęs, kad išlošiu. O, jie ne veltui tą lošimą pavadino „kišenių tuštinimu“.

Konsultantai tuojau suprato, kad tai yra dviejų partnerių nulinės sumos lošimas su trimis galimomis kiekvieno partnerio strategijomis. Jie, susigrūdę krūvon, kažką rašinėjo ant popierinių servetėlių ir galų gale padarė vieningą išvadą, kad Si turėtų skolintis pinigų kur nors kitur, o geriausia — išvis to lošimo nelošti. Jų analizė buvo maždaug šitokia.

Šauklys — pirmasis maksimizuojantis lošėjas, o Si — antrasis minimizuojantis lošėjas. Išlošių matrica šitokia:

Si strategija

$$\begin{array}{c}
 \begin{array}{c} \diamond \\ \clubsuit \\ 2\ \diamond \end{array}
 \begin{pmatrix}
 \begin{array}{ccc} \diamond & \clubsuit & 2\ \clubsuit \end{array} \\
 \begin{array}{ccc} 1 & -1 & -2 \\ -1 & 1 & 1 \\ 2 & -1 & 0 \end{array}
 \end{pmatrix}
 \end{array}$$

Šauklys niekada nesilaikys pirmosios strategijos, nes jis laimės tiek pat arba daugiau, parodydamas būgnų dviakę. Kitaip sakant, išlošis, laikantis trečiosios strategijos, yra didesnis už išlošį, laikantis pirmosios strategijos, arba lygus jam. Todėl faktiškai jis lošia paprastesnį (2×3) lošimą:

---

\* Šiuo atveju tūzas laikomas ekvivalentišku vienai akiai.— *Rus. vert. past.*

$$\begin{array}{ccc}
 & \diamond & \clubsuit & 2\clubsuit \\
 \clubsuit & & & \\
 2\diamond & \left( \begin{array}{ccc} -1 & 1 & 1 \\ 2 & -1 & 0 \end{array} \right)
 \end{array}$$

Lošdamas Si neturėtų laikytis trečiosios strategijos (kryžių dviakė), nes tokį pat arba net geresnį rezultatą jis gauna, laikydamasis antrosios strategijos. Pagaliau lieka tik  $(2 \times 2)$  lošimas:

$$\begin{array}{cc}
 & \diamond & \clubsuit \\
 \clubsuit & & \\
 2\diamond & \left( \begin{array}{cc} -1 & 1 \\ 2 & -1 \end{array} \right)
 \end{array}$$

Sakykime,  $x_2$  ir  $x_3$  yra tikimybės, kad šauklys pasirinks antrąją ir trečiąją strategiją (šiuo atveju  $x_1=0$ ). To lošimo modelis šitoks: rasti maksimalią  $v$  reikšmę, kai

$$-x_2 + 2x_3 \geq v,$$

$$x_2 - x_3 \geq v,$$

$$x_2 + x_3 = 1,$$

$$x_2 \geq 0,$$

$$x_3 \geq 0.$$

Norėdami, kad būtų tik du kintamieji, išreiškiame  $x_3 = 1 - x_2$  ir gauname šiuos apribojimus:

$$-x_2 + 2(1 - x_2) \geq v,$$

$$x_2 - (1 - x_2) \geq v,$$

$$0 \leq x_2 \leq 1,$$

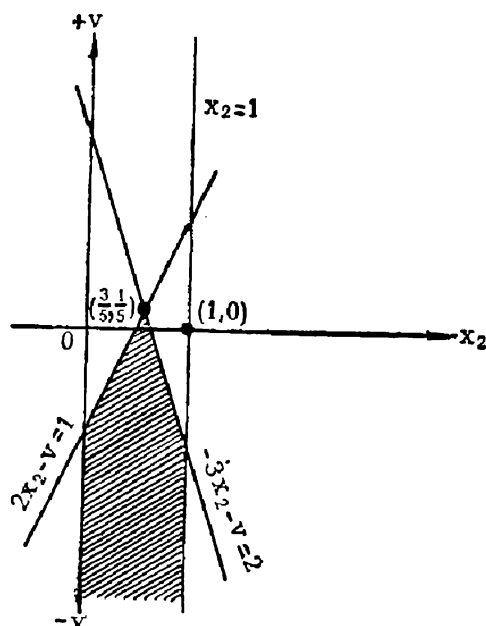
arba

$$-3x_2 - v \geq -2,$$

$$2x_2 - v \geq 1,$$

$$0 \leq x_2 \leq 1.$$

Sprendinių erdvė yra užbrūkšniuota plokštumos  $x_2, v$  sritis.

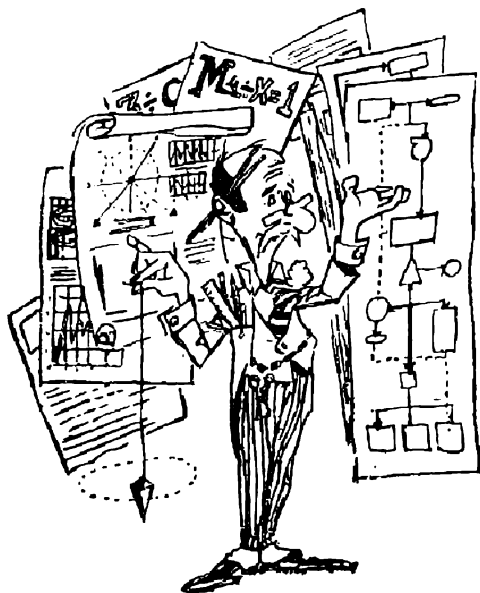


Optimalus taškas bus  $x_2 = \frac{3}{5}$ ,  $v = \frac{1}{5}$ . Vadinasi, lošimo vertė lygi  $\frac{1}{5}$ . Todėl lošimas yra naudingas šaukliui, kai jis niekada nesilaiko pirmosios strategijos, antrąją strategiją pasirenka su tikimybe  $\frac{3}{5}$ , o trečiąją — su tikimybe  $\frac{2}{5}$ , t. y.  $x_1 = 0$ ,  $x_2 = \frac{3}{5}$ ,  $x_3 = \frac{2}{5}$  ir  $v = \frac{1}{5}$ . Tai neteisingas lošimas: šauklys turi pranašumą, o Si tektų dar kartą patvirtinti savo reputaciją. Tačiau ši loginė analizė Si nepadarė nė mažiausio įspūdžio, nes „jeigu jau sekasi, tai sekasi...“

## MATEMATINIS PRIEDAS IR TIESINIO PROGRAMAVIMO TAIKYMŲ SĄRAŠAS

(Kad ir ką įsigytum,  
pagalvok, kas bus toliau,  
ir niekad nesielsi neteisingai.)

Šis priedas padės skaitytojui nutiesti tiltą tarp iki šiol dėstyty klausimų lygio ir specialių tiesinio programavimo darbų lygio. Norint saugiai pereiti tuo tiltu, reikia mo-



kėti žinių muitą, surinktą iš įvairiausių matematikos, skaičiavimo metodų, ekonomikos, valdymo mokslo, inžinerijos ir operacijų tyrimo sričių. Tačiau, norint pakelti užkardą, užtveriančią kelią į tiltą, prie tos medžiagos dar reikia pridėti intuicijos ir kūrybinių gabumų.

### I. Įvadas\*

Programavimo uždavinius sprendžiame, norėdami efektyviai paskirstyti turimus ribotus resursus ir pasiekti pagėidaujamą tikslą. Tokiems uždaviniams būdinga daug

\* Čia pateikiama medžiaga remiasi Garvino [20], Gaso [21], Hedlio [23] ir Vaidos [35] darbais. Skaičiai skliaustuose yra į literatūros sąrašą įtrauktų atitinkamų darbų numeriai.

sprendinių, tenkinančių pagrindines kiekvieno uždavinio sąlygas. Nuo konkretaus tikslo, nurodomo, formuluojant uždavinį, priklauso, kurį atskirą sprendinį laikysime geriausiu uždavinio sprendiniu. Sprendinys, tenkinantis ir uždavinio sąlygas, ir nurodytąjį tikslą, vadinamas *optimaliu sprendiniu*. Tipiškas tokio uždavinio pavyzdys yra uždavinys, kurį sprendžia įmonininkas, norintis taip paskirstyti turimus resursus, kad ne tik patenkintų gamybos poreikius, bet ir gautų maksimalų pelną. Pagrindinės tokio uždavinio sąlygos yra apribojimai, susiję su turimų resursų kiekiu ir gamybos reikalavimais, o jo tikslas — maksimizuoti pajamas.

Čia nagrinėsime tik labai specialų programavimo uždavinių poklasį, kurį sudaro tiesinio programavimo uždaviniai. Iš visų kitų programavimo uždavinių tiesinio programavimo uždaviniai skiriasi tuo, kad, sudarydami jų matematinius modelius, naudojames tik vadinamosiomis „tiesinėmis“ priklausomybėmis. Matematiškai tas priklausomybes reiškiamo lygtimis

$$a_1x_1 + a_2x_2 + \dots + a_jx_j + \dots + a_nx_n = a_0;$$

čia  $a_j$  — žinomi koeficientai, o  $x_j$  — nežinomi kintamieji. Tiesinio programavimo uždavinį formuluojant matematiškai, pateikiama tiesinių lygčių arba nelygybių sistema, išreiškianti uždavinio sąlygas, ir tiesinė funkcija, išreiškianti uždavinio tikslą.

Šiame priede aiškinamos pagrindinės tiesinio programavimo sąvokos, apžvelgiami standartiniai skaičiavimo metodai ir pateikiamas įvairiausių taikymų sąrašas. Kadangi dėl ribotos šios knygos apimties neįmanoma išdėstyti joje teorinių ir matematinių tiesinio programavimo pagrindų, o visiškai aišku, kad, neišmokus to matematikos skyriaus pagrindų, neįmanoma išnaudoti visų jo galimybių, tai skaitytojui reikia susipažinti su specialiais darbais, nurodytais bibliografijoje.

## II. Apibrėžimai ir pagrindinės tiesinio programavimo teoremos

*Tiesinio programavimo uždavinys* formuluojamas šitaip.

Rasti skaičių  $x_1, x_2, \dots, x_n$  aibę, minimizuojančią (arba maksimizuojančią) *tiesinę tikslo funkciją*

$$c_1x_1 + c_2x_2 + \dots + c_jx_j + \dots + c_nx_n \quad (1)$$

ir tenkinančią *tiesinius apribojimus*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1n}x_n &= a_{10}, \\ \dots & \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ij}x_j + \dots + a_{in}x_n &= a_{i0}, \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mj}x_j + \dots + a_{mn}x_n &= a_{m0} \end{aligned} \quad (2)$$

bei neneigiamumo sąlygas

$$\begin{array}{rcl} x_1 & \geq & 0, \\ x_2 & \geq & 0, \\ & \dots & \\ x_j & \geq & 0, \\ & \dots & \\ x_n & \geq & 0. \end{array} \quad (3)$$

Kitaip sakant, reikalaujama rasti neneigiamą tiesinių apribojimų sistemos sprendinį, kuris optimizuoja, t. y. minimizuoja arba maksimizuoja, tiesinę tikslo funkciją. Tokiu atveju  $c_j$  vadinami vertės koeficientais, o  $a_{ij}$  — laisvaisiais nariais. Tiesiniai apribojimai gali būti išreikšti anksčiau pateiktomis lygtimis ir tiesinėmis nelygybėmis

$$a_{i1}x_1 + \dots + a_{in}x_n \leq a_{i0} \quad (4)$$

arba

$$a_{i1}x_1 + \dots + a_{in}x_n \geq a_{i0}. \quad (5)$$

(4) ir (5) pavidalo nelygybes galima pakeisti lygtimis, pridėjus arba atėmus tinkamai parinktą *laisvąjį kintamąjį*. Iš (4) nelygybės gauname, pavyzdžiui, lygtį

$$a_{i1}x_1 + \dots + a_{in}x_n - x_{n+1} = a_{i0}, \quad x_{n+1} \geq 0, \quad (6)$$

o iš (5) — lygtį

$$a_{i1}x_1 + \dots + a_{in}x_n + x_{n+1} = a_{i0}, \quad x_{n+1} \geq 0. \quad (7)$$

Pabrėšime, kad kiekvieną nelygybę atitinka laisvasis kintamasis, susijęs su ta nelygybe. Kiekvienas laisvasis kintamasis yra tos nelygybės kairiosios ir dešinėsios pusių skirtumas.

Kadangi visi tiesinio programavimo uždavinio kintamieji turi būti neneigiami, tai atkreipsime dėmesį, kad

kintamuosius, kurių reikšmės yra ir teigiamos, ir neigiamos, visada galima išreikšti dviejų neneigiamų kintamųjų skirtumu, pavyzdžiui,  $x_i = x'_i - x''_i$ ,  $x'_i \geq 0$ ,  $x''_i \geq 0$ .

Tiesinio programavimo uždavinio tiesiniai apribojimai gali būti labai įvairūs. Skaičiavimui pagrindinius tiesinio programavimo uždavinio apribojimus (2) visada reikia reikšti lygtimis, kurių skaičius ( $m$ ) yra mažesnis už kintamųjų skaičių ( $n$ ). Tuo atveju (2) sistema yra *neapibrėžta* tiesinių lygčių sistema, turinti be galo daug leistinų sprendinių. Kadangi kiekvieną tokią lygtį galima laikyti  $n$ -matės erdvės *hiperplokštumos* lygtimi, tai tiesinės sistemos *sprendinių erdvė*, apskritai kalbant, yra *iškilusis briaunainis*. Tiesinio programavimo skaičiavimo metodais iš visų leistinų sprendinių galima išskirti tą sprendinį, kuris optimizuoja tikslo funkciją. Kadangi sprendinių erdvė gali būti neapibrėžta, tai tikslo funkcijos optimali reikšmė gali būti be galo didelė. Nagrinėsime tik tą atvejį, kai tikslo funkcija minimizuojama, nes maksimizuoti tikslo funkciją — tai tas pat, kas minimizuoti tokią pat funkciją, bet priešingo ženklo.

Vartojant matricinius žymėjimus, uždavinys aprašomas šitaip: minimizuoti  $\mathbf{cX}$ , tenkinant sąlygas

$$\mathbf{AX} = \mathbf{b},$$

$$\mathbf{X} \geq 0;$$

čia  $\mathbf{c} = (c_1, c_2, \dots, c_n)$  — vektorius-eilutė,  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  — vektorius-stulpelis,  $\mathbf{A}$  — koeficientų matrica ( $m \times n$ ),  $\mathbf{b} = (a_{10}, a_{20}, \dots, a_{m0})$  — vektorius-stulpelis,  $\mathbf{0} = (0, 0, \dots, 0)$  — vektorius-stulpelis su  $n$  eilučių. Matricos  $\mathbf{A}$  stulpelius patogiau laikyti  $m$ -matės erdvės taškais; todėl tiesinio programavimo uždavinį galima suformuluoti šitaip: minimizuoti  $\mathbf{cX}$ , tenkinant sąlygas

$$x_1 \mathbf{P}_1 + x_2 \mathbf{P}_2 + \dots + x_n \mathbf{P}_n = \mathbf{P}_0, \quad x_j \geq 0,$$

kai  $\mathbf{P}_j = (a_{1j}, a_{2j}, \dots, a_{mj})$  — vektorius-stulpelis, o  $j = 0, 1, \dots, n$ .

Tiesinio programavimo uždavinio *leistinu sprendiniu* vadinamas vektorius  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , tenkinantis (2) ir (3) sąlygas.

*Pagrindiniu* (2) sistemos *sprendiniu* vadinamas toks sprendinys, kuris gaunamas, kai  $n - m$  kintamųjų laikomi

lygiais nuliui, kiti  $m$  kintamųjų randami iš (2) lygčių sistemos ir, be to, determinantas, sudarytas iš koeficientų, parašytų prie tų  $m$  kintamųjų, nelygus nuliui. Tie  $m$  kintamųjų vadinami „pagrindiniais kintamaisiais“.

*Pagrindiniu leistinu sprendiniu* vadinamas pagrindinis sprendinys, tenkinantis (3) apribojimus, t. y. turintis neneigiamus pagrindinius kintamuosius.

*Neišsigimusiū pagrindiniu leistinu sprendiniu* vadinamas pagrindinis leistinas sprendinys, turintis  $m$  teigiamų  $x_i$ .

*Minimaliu leistinu sprendiniu* vadinamas leistinas sprendinys, minimizuojantis (1) funkciją.

*Bazė* vadinama tiesiškai nepriklausomų vektorių aibė. Tiesinio programavimo uždavinio *leistina bazė* vadinama kvadratinė matrica  $\mathbf{B}$ , sudaryta iš tiesiškai nepriklausomų vektorių, taip parinktų iš stačiakampės matricos  $\mathbf{A} = (\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n)$ , kad sistema  $\mathbf{B}\mathbf{X}_0 = \mathbf{P}_0$  tenkintų sąlygą  $\mathbf{X}_0 = \mathbf{B}^{-1}\mathbf{P}_0 \geq \mathbf{0}$ . Pavyzdžiui, jeigu  $\mathbf{B} = (\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_m)$ , tai gauname  $\mathbf{X}_0 = (x_{10}, x_{20}, \dots, x_{m0}) \geq \mathbf{0}$ . Tokiu atveju tiriamojo uždavinio sprendinys bus vektorius  $\mathbf{X} = (x_{10}, x_{20}, \dots, x_{m0}, 0, \dots, 0)$ , kurio  $n - m$  paskutinių koordinačių lygios nuliui. Rasti leistiną bazę yra tas pats, kas iš neapibrėžtos „stačiakampės“ sistemos pasirinkti apibrėžtą „kvadratinę“ lygčių sistemą, o kintamuosius, neįeinančius į „kvadratinę“ sistemą, pakeisti nuliais.

Iškilioju vektorių  $\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_n$  dariniu vadinamas vektorius

$$\mathbf{U} = \alpha_1 \mathbf{U}_1 + \alpha_2 \mathbf{U}_2 + \dots + \alpha_n \mathbf{U}_n,$$

kai skaliarai  $\alpha_i$  yra neneigiami ir  $\sum \alpha_i = 1$ .

Euklido erdvės taškų poaibis  $\mathbf{C}$  yra *iškilioji aibė* tada ir tik tada, kai, pasirinkus bet kuriuos du poaibio  $\mathbf{C}$  taškus  $\mathbf{U}_1$  ir  $\mathbf{U}_2$ , visi jų iškilieji dariniai

$$\mathbf{U} = \alpha_1 \mathbf{U}_1 + \alpha_2 \mathbf{U}_2 = \alpha \mathbf{U}_1 + (1 - \alpha) \mathbf{U}_2, \quad 0 \leq \alpha \leq 1,$$

priklauso tam pačiam poaibiui  $\mathbf{C}$ . Iškilioji aibė — tai tokia aibė, kuriai priklauso kiekviena atkarpa, jungianti bet kuriuos du tos aibės taškus.

Iškiliosios aibės  $\mathbf{C}$  taškas  $\mathbf{U}$  vadinamas *kraštiniu tašku*, jei  $\mathbf{U}$  negalima išreikšti dviejų skirtingų aibės  $\mathbf{C}$  taškų iškilioju dariniu.

**1 t e o r e m a.** Visų leistinų tiesinio programavimo uždavinio sprendinių aibė yra iškilioji.

2 t e o r e m a. (1) tikslo funkcija įgyja minimalią reikšmę bent viename kraštiniame taške iškiliosios aibės  $C$ , sudarytos iš visų leistinų tiesinio programavimo uždavinio sprendinių. Jeigu minimalią reikšmę ji įgyja keliuose kraštiniuose taškuose, tai ją įgyja ir bet kokiame iškilajame tų taškų darinyje.

3 t e o r e m a. Jeigu vektoriai  $P_1, P_2, \dots, P_k$  ( $k \leq m$ ) yra tiesiškai nepriklausomi ir, be to,

$$x_1 P_1 + x_2 P_2 + \dots + x_k P_k = P_0,$$

o visi  $x_i \geq 0$ , tai taškas  $X = (x_1, x_2, \dots, x_k, 0, \dots, 0)$  yra iškiliosios leistinų sprendinių aibės kraštinis taškas. Čia  $X$  yra  $n$ -matis vektorius, kurio  $n - k$  paskutinių koordinačių lygios nuliui.

4 t e o r e m a. Jeigu  $X = (x_1, x_2, \dots, x_n)$  yra kraštinis aibės  $C$  taškas, tai vektoriai, atitinkantys teigiamus  $x_i$ , yra tiesiškai nepriklausomi. Iš čia išplaukia, kad tokių  $x_i$  yra ne daugiau kaip  $m$ .

5 t e o r e m a. Taškas  $X = (x_1, x_2, \dots, x_n)$  yra kraštinis aibės  $C$  taškas tada ir tik tada, kai teigiami  $x_j$  yra koeficientai prie tiesiškai nepriklausomų vektorių  $P_j$  reiškinyje

$$\sum_{j=1}^n x_j P_j = P_0.$$

6 t e o r e m a. Jeigu egzistuoja bent vienas leistinas sprendinys, tai egzistuoja ir pagrindinis leistinas sprendinys.

7 t e o r e m a. Jeigu tikslo funkcija turi baigtinį minimumą, tai bent vienas optimalus sprendinys yra pagrindinis leistinas sprendinys.

Iš tų teoremų išplaukia, kad, ieškant optimalaus sprendinio, užtenka tirti iškiliosios leistinų sprendinių aibės  $C$  kraštinius taškus. Ką reiškia apibrėžtosios sąvokos geometriškai, buvo aptarta 3 skyriuje.

Jeigu pradinis tiesinio programavimo uždavinys formuluojamas šitaip: minimizuoti  $cX$ , tenkinant sąlygas

$$\begin{aligned} AX &\geq b, \\ X &\geq 0, \end{aligned}$$

tai atitinkamas dualusis uždavinys bus šitoks: maksimizuoti  $\mathbf{Wb}$ , tenkinant sąlygas

$$\mathbf{WA} \leq \mathbf{c},$$

$$\mathbf{W} \geq 0,$$

kai  $\mathbf{W} = (w_1, w_2, \dots, w_n)$  yra vektorius-eilutė, sudaryta iš dualiojo uždavinio kintamųjų.

**Dualumo teorema.** Jeigu vienas uždavinys (pradinis arba dualusis) turi baigtinį optimalų sprendinį, tai antrasis irgi turi baigtinį optimalų sprendinį, o ekstreminės atitinkamų tiesinių funkcijų reikšmės sutampa, t. y.  $\min \mathbf{cX} = \max \mathbf{Wb}$ . Jeigu vienas iš tų uždavinių turi begalinį optimalų sprendinį, tai antrasis visiškai neturi leistinų sprendinių.

Dualumo sąvoka ir dualumo teorema yra labai svarbios tiesinio programavimo teorijoje ir skaičiavimo praktikoje.

### III. Tiesinio programavimo metodai\*

#### 1. Simplekso metodas

Pagrindinis skaičiavimo metodas, pritaikomas *bet kuriam* tiesinio programavimo uždaviniui spręsti, yra *simplekso metodas*. Tuo metodu, radę pirmąjį pagrindinį leistiną sprendinį ir atlikę baigtinį žingsnių skaičių, galime rasti minimalų pagrindinį leistiną sprendinį. Kiekvienu žingsniu, arba *iteracija*, gauname naują pagrindinį leistiną sprendinį, kurį atitinkanti tikslo funkcijos reikšmė yra mažesnė už tikslo funkcijos reikšmę, atitinkančią ankstesnį sprendinį (arba blogiausiu atveju lygi tai reikšmei). Tą procesą tęsiame tol, kol randame minimalų sprendinį su baigtine arba begaline tikslo funkcijos reikšme. Dabar matematiškai aprašysime *standartinį simplekso metodą*. Žodis „simpleksas“ metodo pavadinime atsirado todėl, kad viename iš pirmųjų pavyzdžių, išspręstų tuo metodu, figūravo nelygybė  $x_1 + x_2 + \dots + x_n \leq 1$ , kuria apibrė-

---

\* Smulkesnį 1 ir 2 skyreliuose pateiktos medžiagos išdėstymą skaitytojas gali rasti darbuose, nurodytuose literatūros sąraše, nors jis ir nėra pilnas. Norint suprasti kitus skyrelius, reikia naudotis papildoma literatūra.

žiamas  $n$ -matės erdvės simpleksas (apibendrintas tetraedras), koordinačių ašyse atkertantis vienetines atkarpas.

Sakykime, visos vektoriaus  $\mathbf{P}_0$  koordinatės yra neneigiamos. Jeigu  $i$ -oji koordinatė yra neigiama, tai,  $i$ -osios lygties narius padauginę iš  $-1$ , gausime  $a_{i0} \geq 0$ . Tarkime, kad  $\mathbf{B}_1$  — leistina bazė, t. y. kuris nors pagrindinis leistinas sprendinys randamas iš lygčių sistemos  $\mathbf{B}_1 \mathbf{X}_{01} = \mathbf{P}_0$ ,  $\mathbf{X}_{01} = \mathbf{B}_1^{-1} \mathbf{P}_0 \geq 0$ . Praktiškai  $\mathbf{B}_1$  dažniausiai yra vienetinė  $m$ -osios eilės matrica, o atitinkamą pirmąjį leistiną sprendinį gauname iš karto, nes matrica, atvirkštinė vienetinei matricai, irgi yra vienetinė matrica. Tokiu atveju  $\mathbf{X}_{01} = \mathbf{P}_0$ . Kai uždavinyje nėra tinkamos matricos, iš pradžių galima naudotis *pagalbine vienetine baze*. Tą sprendimo būdą aptarsime vėliau.

Tarkime, kad vienetinė bazė pirmajam leistinam sprendiniui rasti egzistuoja, ir taip sunumeruokime bazės vektorius, kad būtų  $\mathbf{B}_1 = (\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_m)$  (toks žingsnis nėra būtinas, jį atlikome tik tam, kad būtų lengviau samprotauti). Tada simplekso lentelė (skaičiavimo lentelė) bus šitokia:

| Bazė           | $\mathbf{P}_0$ | $\mathbf{P}_1 \dots \mathbf{P}_l \dots \mathbf{P}_m$ | $\mathbf{P}_{m+1} \dots \mathbf{P}_j \dots \mathbf{P}_h \dots \mathbf{P}_n$ |
|----------------|----------------|------------------------------------------------------|-----------------------------------------------------------------------------|
| $\mathbf{P}_1$ | $x_{10}$       | $1 \dots 0 \dots 0$                                  | $x_{1,m+1} \dots x_{1j} \dots x_{1h} \dots x_{1n} \dots$                    |
| ....           | .....          | .....                                                | .....                                                                       |
| $\mathbf{P}_l$ | $x_{l0}$       | $0 \dots 1 \dots 0$                                  | $x_{l,m+1} \dots x_{lj} \dots x_{lh} \dots x_{ln} \dots$                    |
| ....           | .....          | .....                                                | .....                                                                       |
| $\mathbf{P}_m$ | $x_{m0}$       | $0 \dots 0 \dots 1$                                  | $x_{m,m+1} \dots x_{mj} \dots x_{mh} \dots x_{mn} \dots$                    |
|                | $x_{00}$       | $0 \dots 0 \dots 0$                                  | $x_{0,m+1} \dots x_{0j} \dots x_{0h} \dots x_{0n} \dots$                    |

Čia  $x_{ij} = a_{ij}$ , kai  $i = 1, \dots, m$ , o  $j = 0, 1, \dots, n$ ; pagrindinis leistinas sprendinys yra  $\mathbf{X}_{01} = (x_{10}, \dots, x_{l0}, \dots, x_{m0}) = \mathbf{B}_1^{-1} \mathbf{P}_0$ . Apskritai galima apibrėžti  $\mathbf{X}_{j1} = (x_{1j}, x_{2j}, \dots, x_{mj}) = \mathbf{B}_1^{-1} \mathbf{P}_j$ . Tikslo funkcijos reikšmė lygi  $x_{00} = \sum_{i \text{ iš bazės}} c_j x_{i0}$ . Skaičiai  $x_{0j}$ , kai  $j = 1, \dots, n$ , apibrėžiami lygybėmis  $x_{0j} = \sum_{i \text{ iš bazės}} c_i x_{ij} - c_j$ . Tose lygybėse parašytosios sumos vadinamos netiesioginėmis vertėmis ir kartais žy-

mimos  $z_j = \sum_{i \text{ iš bazės}} c_i x_{ij}$ . Atkreipsime dėmesį, kad  $x_{0j} = 0$  visiems  $j$  iš bazės. Iš toliau pateikiamų teoremų matyti, kam reikalingi  $x_{0j}$ .

**1 teorema.** Jeigu nelygybė  $x_{0j} > 0$  yra teisinga, kai  $j$  bet koks, tai galima taip sudaryti leistinų sprendinių aibę, kad su bet kuriuo tos aibės elementu nelygybė  $x'_{00} < x_{00}$  yra teisinga, o apatinis  $< x'_{00}$  rėžis yra baigtinis arba begalinis ( $x_{00}$  yra tikslo funkcijos reikšmė, atitinkanti kurį nors leistinų sprendinių aibės elementą).

I atvejis. Jeigu apatinis rėžis yra baigtinis, tai galima sudaryti naują leistiną sprendinį, turintį  $m$  teigiamų kintamųjų; tą sprendinį atitinkanti tikslo funkcijos reikšmė yra mažesnė už reikšmę, atitinkančią ankstesnį sprendinį, t. y.  $-\infty < x_{00} < x'_{00}$ .

II atvejis. Jeigu apatinis rėžis yra begalinis, tai galima sudaryti naują leistiną sprendinį, turintį  $m+1$  teigiamų kintamųjų; tą sprendinį atitinka kiek norima maža tikslo funkcijos reikšmė\*.

**2 teorema.** Jeigu, imant bet kurį pagrindinį leistiną sprendinį  $X = (x_{10}, x_{20}, \dots, x_{m0})$ , nelygybė  $x_{0j} \leq 0$  yra teisinga visiems  $j = 1, 2, \dots, n$ , tai tas sprendinys yra minimalus leistinas sprendinys.

1 teoremoje tariama, kad uždavinys nėra išsigimęs; kitaip sakant, visi pagrindiniai leistini sprendiniai laikomi griežtai teigiamais (visi  $x_{i0} > 0$ ). Tokia prielaida būtina teoriniu požiūriu, nes, ja remdamiesi, galime įrodyti, kad simplekso metodas po baigtinio žingsnių skaičiaus konverguoja. Jeigu uždavinio pagrindinis leistinas sprendinys yra išsigimęs, tai procesas gali *vykti ciklu*, t. y. po baigtinio žingsnių skaičiaus vėl grįšime prie pradinės bazės, todėl niekada negausime optimalaus sprendinio. Nors pavyzdžių su ciklais žinome, bet jie labai nenatūralūs; dažniausiai procesas konverguoja ir kai išsigimęs, ir kai neišsigimęs uždavinys. Be to, yra specialių skaičiavimo būdų, laiduojančių, kad procesas konverguos bet kokių atveju, nors paprastai jais nesinaudojame.

Ieškant naujo pagrindinio leistino sprendinio, reikia atlikti toliau nurodytus žingsnius. Tais žingsniais vieną

---

\* Kalbama apie mažą ne absoliutinę, bet algebrinę reikšmę.—  
*Rus. vert. past.*

po kito keičiame bazės vektorius, kol pasiekiame tą vietą, kur reikia sustoti.

1. Apskaičiuojame visus  $x_{0j}$ .

2. Patikriname, ar visi  $x_{0j} \leq 0$ , kai  $j=1, 2, \dots, n$ . (Ta nelygybių sistema vadinama optimumo kriterijumi.) Jei-gu taip, tai gautasis sprendinys yra optimalus, ir procesą nutraukiame. Jeigu yra toks  $x_{0j}=0$ , kurį atitinkantis vektorius  $P_j$  nepriklauso optimaliai bazei, tai, įvesdami tą vektorių į bazę, gauname kitą optimalų sprendinį. Jeigu tokio  $x_{0j}$  nėra, tai į bazę įvedame vektorių  $P_k$ , kuris atitinka  $x_{0k} = \max x_{0j} > 0$ . Kai tokių vektorių yra ne vienas, pasirenkame bet kurį iš jų.

3. Kad naujasis sprendinys būtų leistinas, iš bazės reikia pašalinti vektorių  $P_l$ , kuris atitinka

$$\frac{x_{l0}}{x_{lk}} = \min_{x_{lk} > 0} \frac{x_{l0}}{x_{lk}},$$

Jeigu tokių vektorių yra keletas, tai iš jų pasirenkame bet kurį. Kai visi  $x_{lk} \leq 0$ , tai uždavinys turi begalinį optimalų sprendinį, todėl procesą nutraukiame. Jeigu santykis  $\frac{x_{l0}}{x_{lk}}$  yra lygus nuliui (išsigimęs atvejis, kai  $x_{l0}=0$ ), tai tikslo funkcijos reikšmė, atitinkanti naująjį sprendinį, sutampa su buvusiąja reikšme. Elementas  $x_{lk}$  vadinamas *atraminiu elementu*.

4. Naująjį sprendinį ir naująją simplekso lentelę sudarome pagal šias formules (Gauso metodas):

$$x'_{ij} = x_{ij} - \frac{x_{ij} x_{lk}}{x_{lk}}, \quad i \neq l;$$

$$x'_{lk} = \frac{x_{lj}}{x_{lk}}.$$

Tos formulės yra teisingos, kai  $i=0, 1, \dots, m$ , o  $j=0, 1, \dots, n$ . Skaičiai  $x'_{ij}$ , kai  $j=0$ , sudaro naują leistiną sprendinį;  $x'_{00}$  yra naujoji tikslo funkcijos reikšmė;  $x'_{0j}$  lygūs naujiesiems netiesioginių ir tiesioginių verčių skirtumams. Tokia transformacija yra ekvivalentiška naujos leistinos bazės  $B_2$  sudarymui; šią bazę atitinka naujas sprendinys — vektorius

$$X_{02} = B_2^{-1} P_0, \text{ o } X_{j2} = B_2^{-1} P_j.$$

Po to analogiškai žingsniai atliekami su naujosios lentelės duomenimis. Atkreipsime dėmesį, kad aprašytoji transformacija pradinės lentelės vienetinę matricą pakeičia esamos bazės atvirkštine matrica.

Jeigu pradinėje uždavinio formuluotėje nėra vienetinės bazės, tai sistemą reikia papildyti *pagalbiniais neneigiamais kintamaisiais*, kiekvienoje lygtyje parašant po vieną tokį kintamąjį. Kartais pilno pagalbinių kintamųjų rinkinio nereikia, nes dalis vienetinių vektorių jau yra pradiniam uždavinyje. Tariaime, kad vertės koeficientai prie pagalbinių kintamųjų yra begaliniai; todėl, kai egzistuoja pradinio uždavinio minimalus leistinas sprendinys, simplekso metodu gautos pagalbinių kintamųjų reikšmės yra lygios nuliui. Jeigu pradinis uždavinys neturi leistinų sprendinių, tai procesas baigiamas optimaliu sprendiniu, kurio pagalbiniai kintamieji yra teigiami. Skaičiavimo lentelę ir veiksmus galima taikyti ir tuo atveju, kai vertės yra begalinės.

Visa tai pailiustruosime pavyzdžiu. Išspręsimė šitokį tiesinio programavimo uždavinį [28]: maksimizuoti  $x_1 + 2x_2$ , kai

$$-x_1 + 3x_2 \leq 10, \quad (a)$$

$$x_1 + x_2 \leq 6, \quad (b)$$

$$x_1 - x_2 \leq 2, \quad (c)$$

$$x_1 + 3x_2 \geq 6, \quad (d)$$

$$2x_1 + x_2 \geq 4, \quad (e)$$

$$x_1 \geq 0,$$

$$x_2 \geq 0.$$

Tą uždavinį pakeisime minimizavimo uždaviniu; be to, įvedę laisvuosius kintamuosius  $x_3, x_4, x_5, x_6$  ir  $x_7$ , vietoj nelygybių gausime lygtis. Kadangi nėra pilnos vienetinės bazės, tai dar prirašysime du pagalbinius kintamuosius  $x_8$  ir  $x_9$ . Tada uždavinį formuluosime šitaip:

$$\begin{aligned} \min = -x_1 - 2x_2 & \quad + \omega x_8 + \omega x_9 \\ -x_1 + 3x_2 + x_3 & = 10, \quad (a') \end{aligned}$$

$$x_1 + x_2 + x_4 = 6, \quad (b')$$

$$x_1 + x_2 + x_5 = 2, \quad (c')$$

$$\begin{array}{rclcl} x_1 + 3x_2 & -x_6 & +x_8 & = 6, & (d') \\ 2x_1 + x_2 & -x_7 & +x_9 & = 4, & (e') \\ & & & & x_j \geq 0; \end{array}$$

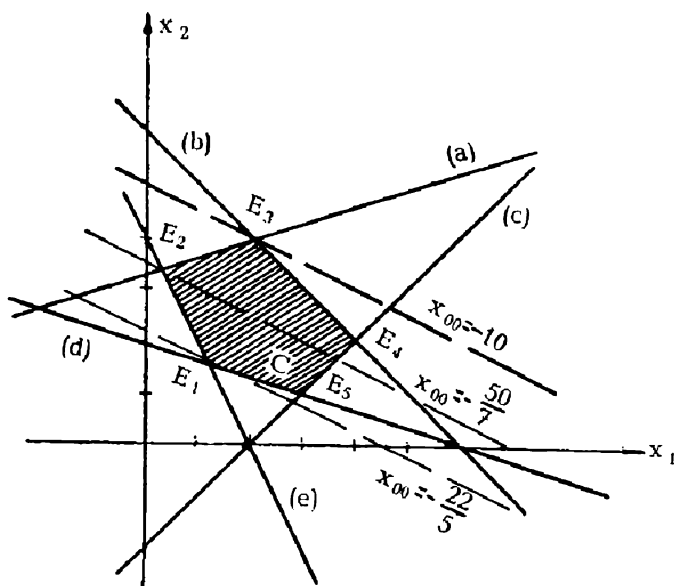
čia  $\omega$  yra begalinė pagalbinė vertė. Pirmasis leistinas pagrindinis sprendinys yra šitoks:  $x_3=10$ ,  $x_4=6$ ,  $x_5=2$ ,  $x_8=6$ ,  $x_9=4$ ; tikslo funkcijos reikšmė lygi  $10\omega$ . Kadangi pagalbinę tikslo funkcijos ir netiesioginės kainos dalį galima atskirti nuo tikrosios tų skaičių dalies, tai lentelėje įrašoma papildoma eilutė. Penkiose lentelėse, pateiktose 123 puslapyje, yra surašytas visas to uždavinio sprendimas. Atkreipkite dėmesį į tai, kad, kol sprendinyje dar yra pagalbinių kintamųjų, į bazę reikia įvedinėti vektorių su netiesioginės vertės maksimalia pagalbine dalimi. Kadangi pagalbinių kintamųjų negalima įvedinėti į bazę pakartotinai, tai juos pašaliname iš lentelių, kai tik jie išnyksta. Elementai, parašyti skritulėliuose, yra atraminiai. Po paskutinės (ketvirtosios) iteracijos gavome optimalų sprendinį:  $x_1=2$ ,  $x_2=4$ ,  $x_5=4$ ,  $x_6=8$ ,  $x_7=4$ ; visi kiti kintamieji lygūs nuliui, o maksimali tikslo funkcijos reikšmė lygi  $+10$ .

|   | Bazė     | $P_0$ | $P_1$         | $P_2$          | $P_3$ | $P_4$ | $P_5$ | $P_6$          | $P_7$ | $P_8$ | $P_9$ |
|---|----------|-------|---------------|----------------|-------|-------|-------|----------------|-------|-------|-------|
| 0 | $P_3$    | 10    | -1            | 3              | 1     |       |       |                |       |       |       |
|   | $P_4$    | 6     | 1             | 1              |       | 1     |       |                |       |       |       |
|   | $P_5$    | 2     | 1             | $\frac{-1}{3}$ |       |       | 1     | -1             |       | 1     |       |
|   | $P_8$    | 6     | 1             | $\frac{3}{1}$  |       |       |       |                |       |       |       |
|   | $P_9$    | 4     | 2             | 1              |       |       |       |                | -1    |       | 1     |
|   |          | 0     | 1             | 2              |       |       |       |                |       |       |       |
|   | $\omega$ | 10    | 3             | 4              |       |       |       | -1             | -1    |       |       |
| 1 | $P_3$    | 4     | -2            |                | 1     |       |       | 1              |       |       |       |
|   | $P_4$    | 4     | $\frac{2}{3}$ |                |       | 1     |       | $\frac{1}{3}$  |       |       |       |
|   | $P_5$    | 4     | $\frac{4}{3}$ |                |       |       | 1     | $-\frac{1}{3}$ |       |       |       |
|   | $P_2$    | 2     | $\frac{1}{3}$ | 1              |       |       |       | $-\frac{1}{3}$ |       |       |       |
|   | $P_9$    | 2     | $\frac{5}{3}$ |                |       |       |       | $\frac{1}{3}$  | -1    |       | 1     |
|   |          | -4    | $\frac{1}{3}$ |                |       |       |       | $\frac{2}{3}$  |       |       |       |
|   | $\omega$ | 2     | $\frac{5}{3}$ |                |       |       |       | $\frac{1}{3}$  | -1    |       |       |

|   |       |         |   |   |        |        |   |                  |        |
|---|-------|---------|---|---|--------|--------|---|------------------|--------|
| 2 | $P_3$ | $32/5$  |   |   | 1      |        |   | $\overline{7/5}$ | $-6/5$ |
|   | $P_4$ | $16/5$  |   |   |        | 1      |   | $1/5$            | $2/5$  |
|   | $P_5$ | $12/5$  |   |   |        |        | 1 | $-3/5$           | $4/5$  |
|   | $P_2$ | $8/5$   |   | 1 |        |        |   | $-2/5$           | $1/5$  |
|   | $P_1$ | $6/5$   | 1 |   |        |        |   | $1/5$            | $-3/5$ |
|   |       | $-22/5$ |   |   |        |        |   | $3/5$            | $1/5$  |
|   | $w$   | 0       |   |   |        |        |   |                  |        |
| 3 | $P_6$ | $32/7$  |   |   | $5/7$  |        |   | 1                | $-6/7$ |
|   | $P_4$ | $16/7$  |   |   | $-1/7$ | 1      |   | $\overline{4/7}$ |        |
|   | $P_5$ | $36/7$  |   |   | $3/7$  |        | 1 | $2/7$            |        |
|   | $P_2$ | $24/7$  |   | 1 | $2/7$  |        |   |                  | $-1/7$ |
|   | $P_1$ | $2/7$   | 1 |   | $-1/7$ |        |   |                  | $-3/7$ |
|   |       | $-50/7$ |   |   | $-3/7$ |        |   |                  | $5/7$  |
| 4 | $P_6$ | 8       |   |   | $1/2$  | $3/2$  |   | 1                |        |
|   | $P_7$ | 4       |   |   | $-1/4$ | $7/4$  |   |                  | 1      |
|   | $P_5$ | 4       |   |   | $1/2$  | $-1/2$ | 1 |                  |        |
|   | $P_2$ | 4       |   | 1 | $1/4$  | $1/4$  |   |                  |        |
|   | $P_1$ | 2       | 1 |   | $-1/4$ | $3/4$  |   |                  |        |
|   |       | -10     |   |   | $-1/4$ | $-5/4$ |   |                  |        |

Pavaizdavę pradines nelygybes dvimatėje erdvėje, kaip nurodyta paveiksle, gauname geometrinių simplekso metodo iliustraciją. Užbrūkšniuotoji sritis  $C$  vaizduoja išskiliają leistinų sprendinių aibę; taškai  $E_i$  yra kraštiniai. Spręsdami šį uždavinį, turėjome atlikti du simplekso metodo žingsnius, kol gavome pirmąjį leistiną pagrindinį sprendinį, atitinkantį kraštinį tašką  $E_1$ ; paskui perėjome prie  $E_2$  ir pagaliau prie optimalaus sprendinio  $E_3$ . Paveiksle pavaizduotos tikslo funkcijos lygio linijos, einančios per tuos kraštinius taškus. Simplekso metodu tam tikra prasme tikslo funkcijos lygio linija perkėlinėjama iš vieno kraštinio taško į arti prie jo esantį kitą kraštinį tašką, kol pasiekiamas optimalus sprendinys. Atkreipsime dėme-

sį, kad optimalių sprendinių būtų be galo daug, jeigu tikslo funkcijos lygio linijos būtų lygiagrečios tiesei, einančiai per taškus  $E_3$  ir  $E_4$ .



## 2. Modifikuotas simplekso metodas

Matricos, atvirkštinės turimai bazei, ieškojimas yra vienas iš svarbiausių tiesinio programavimo skaičiavimų. Kai tik randame tą atvirkštinę matricą, tuoj galime apskaičiuoti visus dydžius, reikalingus eilinei iteracijai. Sakykime,  $B_p$  — leistina  $p$ -osios iteracijos bazė. Tada

$$X_{0p} = B_p^{-1} P_0,$$

$$X_{jp} = B_p^{-1} P_j,$$

$$\pi_p = c_p B_p^{-1},$$

$$\pi_p P_j = c_p B_p^{-1} P_j,$$

$$\pi_p P_0 = c_p X_{0p} = c_p B_p^{-1} P_0;$$

čia  $X_{0p}$  — vektorius, išreiškiantis  $p$ -ąją pagrindinį leistiną sprendinį;  $X_{jp}$  — vektoriai, kuriais naudojantis duotieji

vektoriai  $\mathbf{P}_j$  išreiškiami bazinių vektorių tiesiniais dariniais;  $\mathbf{c}_p$  — vektorius-eilutė, sudarytas iš  $p$ -osios bazės vektorių vertės koeficientų; vektoriaus-eilutės  $\pi_p$  elementai vadinami simplekso daugikliais;  $\pi_p \mathbf{P}_j$  — netiesioginė vektoriaus  $\mathbf{P}_j$  vertė;  $\pi_p \mathbf{P}_0$  — tikslo funkcijos reikšmė, atitinkanti  $p$ -ąją bazę. Naudodamiesi tiksliais atvirkštinės matricos ir simplekso daugiklių išraiškomis, skaičiavimo požiūriu turime keletą pranašumų. Pavyzdžiui, galima sumažinti skaičiavimo operacijų skaičių ir informacijos, kurią būtina užrašyti, atliekant iteracijas, kiekį. Skaičiuojant standartiniu simplekso metodu, reikia užpildyti ir transformuoti visą simplekso lentelę, o naudojantis modifikuoto jo variantu, užtenka parašyti atvirkštinę matricą ir sprendinio vektorių. Pabrėšime, kad, sprendžiant modifikuotu metodu, kiekvienam žingsniui naudojami pradiniai duomenys; dažnai tarp tų duomenų yra daug nulių, todėl sutaupoma skaičiavimui eikvojamo laiko.

Specialus modifikuoto simplekso metodo variantas remiasi tuo faktu, kad matricą, atvirkštinę leistinai bazei, iš pradžių sutampančiai su vienetine matrica, galima išreikšti elementariųjų transformacijų matricų sandauga. Kiekvienoje tokioje matricoje, kuri nuo vienetinės matricos skiriasi tik  $l$ -uoju stulpeliu ( $l$  yra lygus iš bazės pašalinamo vektoriaus numeriui), yra informacija, reikalinga atvirkštinės matricos eilutei rasti. Sakysime,  $p$ -oje iteracijoje turime matricą

$$\mathbf{E}_p^l = \begin{pmatrix} 1 & \dots & y_{1l} & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & y_{ll} & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & y_{ml} & \dots & 1 \end{pmatrix};$$

čia

$$y_{il} = \frac{x_{ik}}{x_{lk}}, \quad i \neq l; \quad y_{ll} = \frac{1}{x_{lk}}.$$

Tada matrica, atvirkštinė  $p$ -ajai bazei, yra šitokia:

$$\mathbf{E}_p^l \mathbf{E}_{p-1}^l \dots \mathbf{E}_2^l \mathbf{E}_1 = \mathbf{B}_p^{-1};$$

$\mathbf{E}_1 = \mathbf{I}$ . Taip glaustai rašant, pakanka užrašinėti tik nedidelį informacijos kiekį. Įrodyta, kad daugeliui tiesinio programavimo uždavinių toks modifikuotas simplekso metodas yra efektyviausias. Iš skaičiavimo praktikos paaiš-

kėjo, kad iteracijų, reikalingų optimaliam sprendiniui rasti, skaičius svyruoja tarp  $m$  ir  $3m$ . Matyt, jis labiau priklauso nuo lygčių skaičiaus, negu nuo kintamųjų. Iteracijų skaičius taip pat priklauso nuo naudojamo algoritmo, nuo metodo, kuriuo ieškoma pirmojo leistino sprendinio, ir nuo kriterijaus, kuriuo remiamasi, pasirenkant vektorių, įvedamą į naują bazę.

### 3. Transporto uždavinio algoritmas

Kadangi koeficientų, apibrėžiančių transporto uždavinį, matrica yra ypatingos struktūros, tai skaičiavimas pagal simplekso metodą šiuo atveju yra labai paprastas. Jei  $m$  yra siuntimo punktų skaičius, o  $n$  — paskirties punktų skaičius, tai kiekvienas leistinas transporto uždavinio sprendinys atitinka trikampę  $(n+m-1)$ -matę bazę. Todėl šiuo atveju iš karto galima rasti sprendinį. Kadangi kiekvienas bazės koeficientas lygus 0 arba 1, tai sprendinys bus išreikštas sveikaisiais skaičiais, kai pradiniai duomenys išreikšti sveikaisiais skaičiais. Simplekso algoritmai transporto uždaviniui spręsti remiasi arba dualiojo uždavinio sprendimu, arba specialiu metodu. Vienas nuo kito jie labiausiai skiriasi būdais netiesioginėms vertėms apskaičiuoti. Dualumo metodas taikomas, sudarant beveik visas mašines programas. Yra ir tokių transporto uždavinio variantų, kurie sprendžiami specialiais skaičiavimo metodais. Iš jų paminėsime uždavinį, kuriame kroviny, pervežamas iš kiekvieno siuntimo punkto į bet kurį paskirties punktą, yra aprėžtas iš viršaus, taip pat apibendrintąjį transporto uždavinį, arba mašinų apkrovimo uždavinį, kurių lygčių koeficientai gali būti nelygūs 0 arba 1.

### 4. Papildomi skaičiavimo būdai

#### *a. Dualus simplekso algoritmas*

Kartais lengviau rasti bazę, tenkinančią optimumo kriterijų (visi  $x_{0j} \leq 0$ ), bet netenkinančią leistinumo kriterijaus (ne visi  $x_{i0} \geq 0$ ). Tokiu atveju galima naudotis dualiojo simplekso algoritmu. Jis skiriasi tik kriterijumi, kuriuo remiamasi, pasirenkant į bazę įvedamą vektorių (kintamąjį) ir iš bazės pašalinamą vektorių. Šiuo atveju

pirmiausia randamas vektorius, kurį reikia pašalinti; jis atitinka:

$$x_{l0} = \min x_{i0} < 0.$$

Naujasis vektorius atitinka indeksą  $k$ , su kuriuo

$$\frac{x_{0k} - c_k}{x_{lk}} = \min_{x_{lj} < 0} \frac{x_{0j} - c_j}{x_{lj}} > 0.$$

Atraminis elementas yra  $x_{lk}$ , o transformacija, kuria pašalinamas vektorius, yra ta pati, kaip ir taikant standartinį simplekso metodą. Po šios transformacijos bent vienas neigiamas kintamasis pasidaro teigiamas, o transformuotieji  $x_{0j}$ , kaip ir anksčiau, tenkina sąlygą  $x_{0j} \leq 0$ , t. y. ta transformacija pagerina sprendinio „leistinumą“, nepažeisdama optimumo kriterijaus. Sudarant *pradinės dualiąsias*, arba *simplekso sudėtines*, mašininės programos, pirmasis simplekso metodas (pradinis metodas) kartais derinamas su dualiuoju metodu. Taikant tuos metodus,  $c_j$  arba  $b_j$  ženklai neapribojami, todėl jais galima gauti pradinį pagrindinį sprendinį, tenkinantį arba leistinumo sąlygas, arba optimumo sąlygas.

### *b. Programavimas sveikaisiais skaičiais*

Daugelio tiesinio programavimo uždavinių sprendinys turi būti išreikštas sveikaisiais skaičiais, pavyzdžiui, kai reikia rasti kurios nors produkcijos sveiką skaičių vienetų. Kitaip, negu transporto uždavinys, simplekso algoritmas nelaiduoja, kad tiesinio programavimo uždavinio sprendinys bendruoju atveju bus išreikštas sveikaisiais skaičiais. Tačiau yra įvairių simplekso metodo variantų, kurie laiduoja, kad optimalus sprendinys, kai jis egzistuoja, bus išreikštas sveikaisiais skaičiais. Tai pasiekama, turimą apribojimų sistemą nuosekliai papildant naujais apribojimais, arba kertančiomis plokštumomis. Naujieji apribojimai taip pakeičia iškiliąją sprendinių aibę, kad optimaliu kraštiniu tašku tampa taškas su sveikomis koordinatėmis. Sudarytos ir atitinkamos mašininės programos, bet dėl to, kad kokiam nors uždaviniui išspręsti reikalingų iteracijų skaičius iš esmės priklauso nuo to uždavinio struktūros ir pradinių duomenų, kol kas tokios programos dar nepritaikytos operacijų tyrimams.

### *c. Aprėžtumo iš viršaus sąlygos*

Tiesinio programavimo uždavinių kintamieji dažnai yra aprėžti iš viršaus, t. y.  $x_j \leq u_j$ . Ir šiuo atveju, nežymiai pakeitus pagrindinį simplekso metodą, galima spręsti uždavinius su tokiais apribojimais, nerašant jų simplekso lentelėje. Tą metodą galima taikyti, kai yra aprėžti visi arba tik kai kurie kintamieji. Uždaviniai, kai  $x_j$  yra aprėžti iš apačios, t. y.  $d_j \leq x_j$ , lengvai įveikiami tiesioginiu keitiniu: tariame, kad  $x_j = d_j + x'_j$ , ir vietoj atitinkamų  $x_j$  parašome  $d_j + x'_j$ .

### *d. Stabilumo analizė*

Stabilumo analizė tiesiniame programavime — tai tyrimas, kokią įtaką optimaliam sprendiniui turi pradinių duomenų keitimas. Pavyzdžiui, mus domina, kokiose ribose galima kaitalioti vertės koeficientus, kad optimalus sprendinys nesikeistų; arba kiek galima keisti koeficientus  $b_i$ , kad sprendinys būtų leistinas; arba, pagaliau, kokią įtaką optimaliam sprendiniui turi skaičių  $a_{ij}$  keitimas. Į visus tuos klausimus galima atsakyti įvairiais būdais, į kuriuos atsižvelgiant, sudaromos atitinkamos mašininės programos. Taip pat sukurti parametrinio programavimo metodai, kai vertės koeficientai ir (arba) laisvieji nariai tiesiškai priklauso nuo kokio nors parametro, pavyzdžiui,  $c_j = d_j + \lambda d'_j$ . Taikant tuos metodus, randami sprendiniai, atitinkantys įvairias parametro reikšmes.

### *e. Skaidymo algoritmas*

Nors teoriškai įmanoma rasti kiekvieno tiesinio programavimo uždavinio sprendinį, bet analitikas greitai suvokia, kad jo siekimai yra apriboti. Svarbiausias apribojimas — apimtis. Beveik visi sunkumai, kylantys, sprendžiant tiesinio programavimo uždavinį, yra vienaip ar kitaip susiję su apimtimi. Tą, be abejo, galima pasakyti apie problemas, susijusias su duomenų rinkimo išlaidomis, matricos sudarymu, kainų apskaičiavimu, tiesinio modelio parinkimu ir t. t.

Daugelio uždavinių apribojimai turi gana daug poabių, kuriuos sudaro lygtys, susijusios su vienu laiko periodu arba tomis pačiomis gamybos galimybėmis. Tie poabiai susieti lygtimis, kurių skaičius nėra didelis. Tokios „sąryšio lygtys“ gali, pavyzdžiui, reikšti bendrą kokio nors produkto poreikį. Tokio tipo uždaviniai tam tikra prasme sudaryti iš ištisos eilės atskirų tiesinio programavimo uždavinių, kurių jungtinis sprendinys dar turi tenkinti kai kuriuos papildomus apribojimus. Jeigu apribojimų sistemą ir tikslo funkciją suskaidysime į blokus, tai gausime

$$\begin{array}{ccccccc}
 \boxed{C_0} & \boxed{C_1} & \boxed{C_2} & \dots & \boxed{C_k} & & \\
 \hline
 \boxed{A_0} & \boxed{A_1} & \boxed{A_2} & \dots & \boxed{A_k} & = & \boxed{b_0} \\
 & & & & & & \\
 & \boxed{B_1} & & & & = & \boxed{b_1} \\
 & & \boxed{B_2} & & & = & \boxed{b_2} \\
 & & & & & & \\
 & & & \dots & & & \\
 & & & & \boxed{B_k} & = & \boxed{b_k}
 \end{array}$$

Šitaip pradinį uždavinį (rasti  $\mathbf{cX}$  minimumą, kai  $\mathbf{AX}=\mathbf{b}$ ,  $\mathbf{X} \geq \mathbf{0}$ ) pakeitėme sistema, sudaryta iš smulkesnių uždavinių: rasti tokius vektorius  $\mathbf{X}_p \geq \mathbf{0}$  ( $p=0, 1, \dots, k$ ), kad

$$\sum_p \mathbf{A}_p \mathbf{X}_p = \mathbf{b}_0, \quad (8)$$

$$\mathbf{B}_p \mathbf{X}_p = \mathbf{b}_p, \quad (9)$$

o

$$\sum_p \mathbf{C}_p \mathbf{X}_p \quad (10)$$

pasiektų minimumą; čia  $\mathbf{A}_p$  yra matrica ( $m_0 \times n_p$ ),  $\mathbf{B}_p$  — matrica ( $m_0 \times n_p$ ),  $\mathbf{C}_p$  —  $n_p$ -matis vektorius,  $\mathbf{b}_0$  —  $m_0$ -matis vektorius,  $\mathbf{b}_p$  —  $m_p$ -matis vektorius,  $\mathbf{X}_p$  —  $n_p$ -matis vektorius, sudarytas iš nežinomųjų. Kaip matome, tame uždavi-

nyje yra  $\sum_p m_p$  apribojimų ir  $\sum_p n_p$  kintamųjų. Dabar tarkime, kad tas uždavinys turi baigtinį optimumą.

Sakykime, kiekvieną  $p$  atitinka iškilioji aibė  $S_p$ , sudaryta iš smulkesnių uždavinių  $B_p X_p = b_p$ ,  $X_p \geq 0$  sprendinių. Tada pradinio uždavinio sprendinys yra tam tikras iškilusis darinys, sudarytas iš sprendinių, priklausančių  $S_p$ , tenkinantis sąryšio  $\sum_p A_p X_p = b_0$  apribojimus ir mini-

mizuojantis reiškinį  $\sum_p C_p X_p$ . Nors ir manoma, kad dėl tos grynos skaidymo idėjos gali kilti naujų sunkumų, bet ji duoda šokią tokią ekonomiją. Vietoj sudėtingo uždavinio su  $\sum_p m_p$  apribojimų, galima nagrinėti optimizavimo uždavinį su  $m_0 + k$  apribojimų, prieš tai išsprendus  $k(m_p \times n_p)$  smulkesnių uždavinių.

Išnagrinėsime naują uždavinį, vadinamą ekstremine programa, arba pagrindiniu uždaviniu. Imkime kurį nors uždavinio  $X_p \geq 0$ ,  $B_p X_p = b_p$  ( $p$  fiksuotas) iškiliosios sprendinių aibės  $S_p$  kraštinį tašką  $X_{pj}$ . Kiekvienam tokiam kraštiniam taškui  $X_{pj}$  apskaičiuosime

$$P_{pj} = A_p X_{pj} \text{ ir } f_{pj} = C_p X_{pj}.$$

Pagal ekstreminę programą reikia rasti tokius skaičius  $\lambda_{pj} \geq 0$ , kurie tenkina lygtis

$$A_0 X_0 + \sum_p \sum_j P_{pj} \lambda_{pj} = b_0, \quad (11)$$

$$\sum_j \lambda_{pj} = 1 \quad (p = 1, 2, \dots, k) \quad (12)$$

ir minimizuoja reiškinį

$$C_0 X_0 + \sum_p \sum_j f_{pj} \lambda_{pj}. \quad (13)$$

Kaip rašoma viename Dancigo ir Volfo darbe, „ekstreminio uždavinio ir pradinio uždavinio sąryšis pasireiškia tuo, kad bet kuris aibės  $S_p$  taškas (pagal prielaidą ši aibė yra aprėžtas ir iškilusis briaunainis) gali būti išreikštas tiesiniu tos aibės kraštinių taškų dariniu, t. y. reiškinium  $\sum_j X_{pj} \lambda_{pj}$ , kurio kintamieji  $\lambda_{pj}$  tenkina (12) lygtį, o (11) ir (13) reiškiniai visiškai sutampa su (8) ir (10) reiškiniais, parašytais su kintamaisiais  $\lambda_{pj}$ .“

Kaip jau nurodėme, tokiu būdu sudarome uždavinį, turintį tik  $m_0 + k$  apribojimų ( $m_0$  (11) sąryšio apribojimų ir  $k$  (12) apribojimų). Tačiau kintamųjų skaičius didėja ir pasidaro lygus bendram iškiliosios aibės  $S_p$  kraštinių taškų skaičiui. Ekonomija, taikant skaidymo metodą, pasireiškia tuo, kad tenka nagrinėti tik nedidelį tų bendrų taškų skaičių, be to, tiksliai išreikšti reikia tik tiriamus taškus ir tik tada, kai tai yra būtina.

#### IV. Tiesinio programavimo taikymų sąrašas

Šiame skyrelyje apžvelgiama 1—4 skyriuose pateikta medžiaga, formaliau aprašomi kai kurie svarbūs tiesinio programavimo taikymai, taip pat, kur įmanoma, sudaromi tipiniai matematiniai modeliai.

Būtina pabrėžti, kad tie modeliai (nors jie savaime yra svarbūs) yra tik išeities pozicijos arba pagrindiniai modeliai, formuluojant panašius uždavinius. Ir skaitytojui gali pasitaikyti uždavinys, panašus į kurį nors iš nurodytųjų modelių. Tada, atitinkamai modifikavus pagrindinį modelį, gal būt, pavyks gauti neblogą pirmąjį priartėjimą prie realios situacijos. Šiuo atžvilgiu puikus pavyzdys yra transporto uždavinys, kai reikia derinti gamybos ir sandėliavimo modelius, norint pavaizduoti tikrąją padėtį. Nereikia uždavinio dirbtinai sprauti į kokio nors modelio rėmus, bet kartu reikia atsiminti, kad jau pats pagrindinio modelio rinkimasis gali sukelti mūsų nuomonės apie tai, kas vyksta tikrovėje, evoliuciją.

##### 1. Transporto uždaviniai

###### a. Pagrindinis transporto uždavinys

Iš  $m$  punktų reikia išsiųsti po  $a_1, a_2, \dots, a_m$  vienetų vienos rūšies produkto ir nugabenti į  $n$  paskirties punktų atitinkamai po  $b_1, b_2, \dots, b_n$  vienetų to produkto. Produkto vieneto pervežimo iš punkto  $i$  į punktą  $j$  kaina  $c_{ij}$  yra žinoma, imant bet kurią kombinaciją  $(i, j)$ . Reikia rasti, kokį produkto kiekį  $x_{ij}$  reikia gabenti kiekvienu keliu  $(i, j)$ , kad bendros pervežimo išlaidos būtų minimalios. Skaičiai  $a_i$  reiškia turimuosius, o  $b_j$  — reikalaujamuosius produkto

kiekius. Siekdami matematinio modelio turiningumo, tar-sime, kad yra teisinga lygybė  $\sum_i a_i = \sum_j b_j$ , t. y. kad turi-mojo produkto kiekis yra lygus reikalaujamam kiekiui. Kai kuriuose konkrečiuose taikymuose šio apribojimo gali ir nebūti. Jeigu  $\sum_i a_i > \sum_j b_j$ , tai uždavinys papildomas fik-tyviu paskirties punktu (pavyzdžiui, perteklius kaupiamas sandėliuose), kuriame laukiama  $\sum_i a_i - \sum_j b_j$  vienetų pro-dukto. Kai  $\sum_i a_i < \sum_j b_j$ , uždavinys papildomas fiktyviu siuntimo punktu (pavyzdžiui, perkama iš konkurento), ku-riame yra  $\sum_j b_j - \sum_i a_i$  vienetų produkto. Sakykime,  $x_{ij} \geq 0$  — nežinomas kiekis, kurį reikia nugabenti iš punkto  $i$  į punktą  $j$  (t. y.  $x_{ij}$  — kintamųjų aibė). Tada tiesinio programavimo modelis, kai  $m=2$ ,  $n=3$ , yra šitoks: mi-nimizuoti

$$\begin{array}{l}
 c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23}, \\
 \text{kai} \\
 \begin{array}{llll}
 x_{11} + x_{12} + x_{13} & & & = a_1, \\
 & x_{21} + x_{22} + x_{23} & & = a_2, \\
 x_{11} & + x_{21} & & = b_1, \\
 & x_{12} & + x_{22} & = b_2, \\
 & & x_{13} & + x_{23} & = b_3, \\
 & & & & x_{ij} \geq 0.
 \end{array}
 \end{array}$$

Kadangi bet kurią čia parašytą lygtį galima gauti iš  $(m+n-1)$  likusių, tai sistemos rangas yra lygus  $(m+n-1)$ . Kiekviena leistina bazė yra  $(m+n-1)$ -osios eilės, o atitinkama matrica — trikampė. Aprašytasis modelis yra plačiai taikomas; siuntimo ir paskirties punktai labai įvai-riai interpretuojami (pavyzdžiui, sandėliai, parduotuvės, uostai), o vertės koeficientai yra atstumai, laikas, dole-riai ir t. t. Tą modelį galima modifikuoti, pavyzdžiui, no-rint gauti informacijos, padedančios iš kelių numatomų naujų siuntimo punktų pasirinkti vieną, geriausią trans-porto kainų atžvilgiu. Tokiu atveju tenka spręsti aibę už-davinių (po vieną kiekvienai kombinacijai), kurių duo-menys atspindi numatomą mokestį už pervežimą iš nau-jojo siuntimo punkto į senuosius paskirties punktus, taip

pat numatomą produkto kiekį naujajame punkte ir reikalaujamus kiekius senuosiuose punktuose. Panaši analizė gali padėti, pasirenkant naują paskirties punktą, taip pat apsisprendžiant, kurį siuntimo arba paskirties punktą reikia uždaryti.

### *b. Tranzito uždavinys*

Tai transporto uždavinys, kuriame siuntimo ir paskirties punktai gali pasidaryti tarpiniais punktais, per kuriuos gabenamos prekės į galutinį paskirties punktą. Šį uždavinį galima pakeisti pagrindiniu transporto uždaviniu ir spręsti metodais, taikomais tam uždaviniui.

### *c. Ribotų galimybių transporto uždavinys*

Tai pagrindinis transporto uždavinys, tik pervežamo produkto kiekis yra aprėžtas iš viršaus, t. y.  $x_{ij} \leq u_{ij}$ . Tokį uždavinį galima išspręsti, truputį modifikavus tą metodą, kuriuo buvo sprendžiamas pagrindinis uždavinys. Tam uždaviniui taikomas ir specialus pradinis dualusis algoritmas.

### *d. Apibendrintas transporto uždavinys ir staklių skirstymo uždavinys*

Šiuo atveju uždavinys formuluojamas šitaip: minimizuoti

$$\sum_i \sum_j c_{ij} x_{ij},$$

kai

$$\sum_{j=1}^n a_{ij} x_{ij} = a_i \quad (i=1, \dots, m), \quad (14)$$

$$\sum_{i=1}^m b_{ij} x_{ij} = b_j \quad (j=1, \dots, n), \quad (15)$$

$$x_{ij} \geq 0.$$

Pagal šį modelį sprendžiami transporto ir staklių skirstymo uždaviniai. Pastaruoju atveju  $b_{ij}=1$ , o (14) lygtys pakeičiamos nelygybėmis ( $\leq$ );  $a_{ij}$  reiškia laiką, per kurį  $i$ -osiomis staklėmis pagaminamas vienas  $j$ -ojo produkto

vienetas;  $x_{ij}$  —  $j$ -ojo produkto, gaminamo  $i$ -osiomis staklėmis, vienetų skaičius;  $b_j$  — būtinų pagaminti  $j$ -ojo produkto vienetų skaičius;  $c_{ij}$  —  $j$ -ojo produkto vieneto gamybos  $i$ -osiomis staklėmis kaina.

### *e. Daugiamatis transporto uždavinys*

Sio uždavinio modelis gali būti dviejų tipų.

#### 1. Minimizuoti

$$\sum_i \sum_j \sum_k c_{ijk} x_{ijk},$$

kai

$$\sum_i x_{ijk} = a_{jk},$$

$$\sum_j x_{ijk} = b_{ik},$$

$$\sum_k x_{ijk} = d_{ij},$$

$$x_{ijk} \geq 0.$$

#### 2. Minimizuoti

$$\sum_i \sum_j \sum_k c_{ijk} x_{ijk},$$

kai

$$\sum_j \sum_k x_{ijk} = a_i,$$

$$\sum_i \sum_k x_{ijk} = b_j,$$

$$\sum_i \sum_j x_{ijk} = d_k,$$

$$x_{ijk} \geq 0.$$

### **2. Skirstymo arba paskyrimo uždaviniai**

Sio tipo uždaviniuose yra tam tikras skaičius individų, mašinų ir t. t., kuriuos reikia paskirstyti, atliekant kokius nors darbus. Kiekvieną individą  $i$  atitinka skaičius  $c_{ij}$ , kuriuo matuojamas jo efektyvumas, atliekant darbą  $j$ . Individas gali būti paskirtas tik į vieną darbą. Jeigu  $x_{ij}$  reiškia  $i$ -ojo individo paskyrimą į  $j$ -ąjį darbą, tai tiesinio programavimo uždavinys formuluojamas šitaip: maksimizuoti

$$\sum_i \sum_j c_{ij} x_{ij},$$

kai

$$\sum_j x_{ij} = a_i \quad (i=1, \dots, m),$$
$$\sum_i x_{ij} = b_j \quad (j=1, \dots, n),$$
$$x_{ij} \geq 0;$$

čia  $a_i$  — turimų  $i$ -ojo tipo individų kiekis, o  $b_j$  — turimų  $j$ -ojo tipo darbų kiekis. Tariaama, kad  $\sum a_i = \sum b_j$ . Labai dažnai visi  $a_i$  ir visi  $b_j$  yra lygūs 1, o  $m=n$ . Matematinio požiūriu skirstymo uždavinys yra analogiškas transporto uždaviniui.

### 3. Tanklaivių uždavinys

Uostuose  $i=1, 2, \dots, m$  pakraunami tanklaiviai ir siunčiami į paskirties uostus  $j=1, 2, \dots, n$ . Žinome, per kiek laiko uoste  $i$  pakraunamas tanklaivis, vykstantis į uostą  $j$ , taip pat, kiek laiko trunka kelionė iš vieno uosto į kitą. (Tariame, kad visi tanklaiviai yra vienodi, o krovinyš išreiškiamas tanklaivio talpos vienetais.) Nugabenęs krovinį, tanklaivis gali plaukti į bet kurį pakrovimo uostą. Reikia sudaryti tokį kiekvieno tanklaivio maršrutą, kad bendrasis tanklaivių skaičius būtų minimalus.

### 4. Tinklų srauto uždaviniai

Duotas koks nors tinklas (plentai, geležinkeliai, vamzdynas ir t. t.), kuriame yra vienintelis šaltinis (siuntimo punktas), vienintelė santaka (paskirties punktas) ir tarpiniai mazgai (persiuntimo punktai). Sakykime,  $x_{ij}$  yra srautas iš to tinklo  $i$ -ojo punkto į  $j$ -ąjį punktą. Punktas gali būti arba šaltinis, arba santaka, arba tarpinis mazgas. Tariaame, kad tinklas yra orientuotas ir kiekvieno kelio  $(i, j)$  laidumas yra aprėžtas  $f_{ij} \geq 0$ . Siekiame taip paskirstyti srautą tinkle, kad bendrasis srautas  $f$  iš šaltinio į santaką būtų maksimalus.

Bendruoju atveju, kai tinkle yra šaltinis  $i=0$  ir santaka  $i=m$ , uždavinį formuluojuame šitaip: maksimizuoti  $f$ , kai

$$\sum_j (x_{0j} - x_{j0}) = f, \quad (16)$$

$$\sum_i (x_{ij} - x_{ji}) = 0, \quad i \neq 0, m, \quad (17)$$

$$\sum_j (x_{mj} - x_{jm}) = -f, \quad (18)$$

$$0 \leq x_{ij} \leq f_{ij}. \quad (19)$$

(17) lygybė išreiškia srauto nepertraukiamumo sąlygą tarpiniuose mazguose, t. y., kas įteka į kurį nors mazgą, tas turi ir ištekti iš jo, o sumuojama pagal tuos  $j$ , kurie tiesiogiai susieti keliais su punktu  $i$ . Tikslą funkcija maksimizuoja srautą iš šaltinio arba srautą į santaką — (16) ir (18).

### 5. Kontraktų sudarymas

Kai valstybinei įstaigai reikia užpirkti kokių nors prekių, ji turi kreiptis į firmas, gaminančias tas prekes, ir sudaryti kontraktus. Kiekviena firma pateikia pasiūlymus, kuriuose atsispindi siekimas gauti pelno, spėliojimai konkurentų pasiūlymų atžvilgiu ir jos pačios keliama reikalavimai. Pirkėjas siekia sudaryti tokius kontraktus, kad bendroji sąnaudų suma būtų minimali. Tokio tipo uždavinius galima pakeisti transporto uždavinių seką ir spręsti, naudojantis tų uždavinių sprendimo algoritmu.

### 6. Veiklos analizės uždaviniai

Koks nors įmonininkas savo žinioje turi tam tikrą įvairių resursų kiekį. Tie resursai, pavyzdžiui, žaliava, darbas ir įranga, naudojami kokioms nors prekėms arba prekių deriniams gaminti. Įmonininkas žino, kiek reikia resurso  $i$  vienam prekės  $j$  vienetui pagaminti. Taip pat jis žino pajamas, kurias gaus, pardavęs vieną pagamintos prekės  $j$  vienetą. Įmonininkas siekia pagaminti tokių prekių derinius, kurie maksimizuos jo bendrąsias pajamas. Vartosime šiuos žymėjimus:

- $m$  — resursų kiekis,
- $n$  — prekių skaičius,
- $a_{ij}$  — vienam prekės  $j$  vienetui pagaminti sunaudojamo resurso  $i$  vienetų skaičius;
- $b_i$  — maksimalus resurso  $i$  turimų vienetų skaičius;
- $c_j$  — pajamos, gaunamos, pardavus vieną pagamintos prekės  $j$  vienetą;
- $x_j$  — prekės  $j$  gamybos lygis (pagamintas kiekis).

Bendras suvartoto  $i$ -ojo resurso kiekis išreiškiamas tiesiniu reiškiniu

Kadangi bendras kiekis negali būti didesnis už maksimalų turimų  $i$ -ojo resurso vienetų skaičių, tai kiekvieną  $i$  atitinka tiesinė nelygybė

Neigiamų  $x_j$  neįmanoma tinkamai interpretuoti, todėl reikia, kad visi  $x_j$  tenkintų nelygybę  $x_j \geq 0$ . Pajamos, gaunamos, pardavus  $x_j$ -osios prekės vienetų, lygios  $c_j x_j$ . Uždavinys formuluojamas šitaip: maksimizuoti pajamų funkciją

kai

$$x_n \geq 0.$$

Tie uždaviniai išskyla ekonomikoje, nagrinėjant firmų teoriją ir atliekant tarpšakinę analizę (sąnaudos—išeiga).

Sakykime, kokių nors maisto produktų sudėtis yra žinoma. Pavyzdžiui, žinome, kiek miligramų fosforo arba geležies yra kiekvieno aptariamojo produkto uncijoje. Taip pat žinome, kiek mažiausiai per dieną reikia kiekvienos medžiagos, būtinos gyvybinei veiklai palaikyti, ir kiek kainuoja kiekvieno maisto produkto uncija. Reikia sudaryti

pigiausią dietą, tenkinančią minimalius baltymų, riebalų, vitaminų ir t. t. poreikius. Sakykime,

$m$  — būtinų medžiagų skaičius;

$n$  — maisto produktų skaičius;

$a_{ij}$  — vienoje  $j$ -ojo maisto produkto uncijoje esančios  $i$ -osios medžiagos miligramų skaičius;

$b_i$  — minimalus  $i$ -osios medžiagos kiekis, būtinas normaliai organizmo veiklai;

$c_j$  —  $j$ -ojo maisto produkto vienos uncijos kaina;

$x_j$  — skaičius, reiškiantis, kiek uncijų  $j$ -ojo produkto reikia nupirkti ( $x_j \geq 0$ ).

Bendras  $i$ -osios medžiagos, esančios visame pirkinyje, kiekis išreiškiamas suma

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n.$$

Tas kiekis turi būti didesnis už minimalią  $i$ -osios medžiagos dienos normą arba lygus jai. Tiriamasis uždavinys formuluojamas šitaip: minimizuoti kainos funkciją

$$c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

kai

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2,$$

$$\dots \dots \dots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$\dots \dots \dots$$

$$x_n \geq 0.$$

Nors tokia uždavinio formuluo­­tė ir abejotina, kalbant apie žmonių dietas, bet ji yra visiškai tinkama, sudarinėjant stambiųjų raguočių ir viščiukų racionus.

### 8. Mišinių sudarymo uždavinys

Tokius uždavinius tenka spręsti, kai reikia sudaryti vieną ar kelis tam tikrų produktų mišinius. Šiuo atveju yra žinomi turimų žaliavos rūšių kiekio ir gaminamų produktų kiekio bei kokybės apribojimai. Paprastai iš turimų

žaliavos rūšių galima sudaryti be galo daug visokių mišinių, tenkinančių įvairiausius apribojimus. Reikia sudaryti mišinį, optimizuojantį nurodytąją tikslo funkciją. Mišinių uždavinių pasitaiko naftos pramonėje, dažų, plieno ir t. t. gamyboje. Aprašant atitinkamą matematinį modelį, reikėtų gilintis į konkrečių gamybos procesų detales, o tai netelpa šios knygos rėmuose. Skaitytojui, kuris tuo domisi, patariame skaityti literatūrą, nurodytą bibliografijoje.

## 9. Gamybos planavimas

Įmonininkas žino, kad per  $n$  artimiausių mėnesių reikia pagaminti  $r_i$  ( $i=1, 2, \dots, n$ ) vienetų kokios nors prekės. Tą kiekį galima pagaminti arba dirbant normaliai ir per mėnesį pagaminant ne daugiau kaip  $a_i$  vienetų, arba „šturmuojant“ ir per mėnesį pagaminant ne daugiau kaip  $b_i$  vienetų. Vieno prekės vieneto gamybos kaina  $i$ -ąjį mėnesį lygi  $c_i$ , kai dirbama reguliariai, arba  $d_i$ , kai gaminama forsuotai. Dėl kainos svyravimo laikui bėgant, taip pat dėl ribotų gamybos galimybių kartais naudingiau pagaminti tos prekės atsargą, nelaukiant, kol jos iš tikrųjų prireiks. Sakykime, prekės vieneto laikymas sandėlyje kainuoja  $s_i$  dolerių per mėnesį. Reikia taip planuoti gamybą, kad bendroji gamybos ir sandėliavimo sąnaudų suma būtų minimali. Nors tą uždavinį galima formuluoti kaip standartinį tiesinio programavimo, bet taip pat galima jį nusakyti transporto uždavinio terminais ir spręsti pagal to uždavinio algoritmą.

## 10. Išlyginamoji gamybos schema

Įmonininkas turi sudaryti  $n$  artimiausių mėnesių kokios nors prekės gamybos planą. Nors to gaminio paklausa keičiasi, bet vis tiek reikia nuolat tenkinti ir anksčiau gautus, ir būsimus mėnesio užsakymus. Norint išpildyti individualius užsakymus, per mėnesį reikia pagaminti visą reikalaujamą kiekį arba tik jo dalį, trūkumą padengiant ankstesnių mėnesių produkcijos pertekliumi.

Apskritai kalbant, kiekvienas planavimo uždavinys turi daug įvairių sprendinių, tenkinančių nurodytuosius apribojimus. Pavyzdžiui, kiekvieną mėnesį įmonininkas gali

gaminti kaip tik tą produkcijos kiekį, kuris, jo nuomone, bus užsakytas. Tačiau toks planas nėra efektyvus, nes, jo laikantis, susidaro daug nuostolių. Kita vertus, įmonininkas, sumažėjęs gaminių paklausai, gali nesiaurinti gamybos ir susidariusį perteklių panaudoti, padidėjęs paklausai. Vadinasi, gamyba gali būti visiškai stabili. Tačiau toks klausimo sprendimas gali pasirodyti netinkamas, jei gaminių sandėliavimo mėnesinės išlaidos yra palyginti didelės. Reikia taip planuoti gamybą, kad išlaidų, susidarančių dėl gamybos fliktuacijos ir gaminių saugojimo, suma būtų minimali.

Sakykime,

$x_t$  — per  $t$ -ąjį mėnesį pagamintos produkcijos kiekis;

$r_t$  —  $t$ -ąjį mėnesį reikalaujamos produkcijos kiekis;

$s_t$  —  $t$ -ąjį mėnesį sandėlyje saugotos produkcijos kiekis.

Aptariamasis uždavinys formuluojamas šitaip: minimizuoti

$$a \sum_t y_t + b \sum_t s_t,$$

kai

$$x_t + s_{t-1} - s_t = r_t, \quad (20)$$

$$x_t - x_{t-1} - y_t + z_t = 0; \quad (21)$$

čia  $(x_t, s_t, y_t, z_t) \geq 0$ . (20) lygybė reiškia, kad  $t$ -ąjį mėnesį pagamintos ir praėjusį mėnesį sandėliuotos produkcijos kiekis yra lygus  $t$ -ąjį mėnesį pareikalautos ir tą mėnesį sandėliuotos produkcijos kiekiui. (21) lygybė reiškia, kad  $t$ -ojo mėnesio ir  $(t-1)$ -ojo mėnesio produkcijos kiekių skirtumą galima laikyti dviejų neneigiamų skaičių  $y_t$  ir  $z_t$  skirtumu;  $y_t$  yra produkcijos padidėjimas, o  $z_t$  — jos sumažėjimas.  $a$  yra produkcijos padidėjimo vienu vienetu kaina, o  $b$  — produkcijos vieneto saugojimo sandėlyje vieną mėnesį kaina.

## 11. Nuostolių uždavinys

Popieriaus fabrikai gamina tam tikro standartinio pločio rulonus. Užsakovams reikia įvairaus pločio rulonų, todėl standartinius rulonus tenka piaustyti. Piaustymo procesas, apskritai kalbant, yra susijęs su tam tikrais nuostoliais. Rulonus reikia taip piaustyti, kad būtų ne tik išpildomi užsakovų reikalavimai, bet ir minimizuojami bend-

rieji nuostoliai. Tą pačią schemą galima taikyti analogiškiems gamybos procesams, kai, piaustant standartinius rulonus, lapus ir pan., atsiranda nuostolių.

## 12. Tiekėjo uždavinys

Kulinarinių gaminių tiekėjas žino, kad per  $n$  artimiausių dienų kasdien jam reikės po  $r_j$  ( $j=1, 2, \dots, n$ ) švarių servetėlių. Skalbiama dažniausiai  $p$  dienų, t. y. jeigu nešvari servetėlė, naudota  $j$ -ąją dieną, tuoj pat siunčiama į skalbyklą, tai  $(j+p)$ -ąją dieną ją vėl galima naudoti. Be to, skalbykloje ir skubiai aptarnaujama; tada švarios servetėlės gaunamos po  $q$  dienų ( $q < p$ ,  $p$  ir  $q$  — sveikieji skaičiai). Neturėdamas po ranka arba skalbykloje reikalingo servetėlių skaičiaus, tiekėjas iš pradžių perka jas, mokėdamas po  $a$  centų už kiekvieną. Paprastas vienos servetėlės skalbimas kainuoja  $b$  centų, o skubus —  $c$  centų. Ką turi daryti tiekėjas, kad patenkintų savo poreikius ir minimizuotų išlaidas per  $n$  nurodytųjų dienų?

## 13. Komivojažieriaus uždavinys

Reikia rasti trumpiausią kelią, kuriuo vykdamas komivojažierius galėtų aplankyti tam tikrą miestų grupę ir grįžti atgal, į išvykimo punktą. Formuluojuojant šį tiesinio programavimo uždavinį, reikalaujama, kad kintamųjų reikšmės būtų sveikieji skaičiai.

## 14. Sandėlio uždavinys

Sakykime, turime tam tikros talpos sandėlį, o jame — pradinį kiekį kokio nors produkto, kurio kaina ir vertė svyruoja priklausomai nuo sezono. Be to, žinome, kiek laiko praeina nuo produkto pirkimo iki jo gavimo. Kokios pirkimo schemos reikia laikytis, norint maksimizuoti pajamas per nurodytą laikotarpį?

## **15. Tinklų planavimas**

Daugeliui projektų būdinga tai, kad visas darbas turi būti atliekamas tam tikra tvarka; pavyzdžiui, statyboje iš pradžių reikia padaryti klojinį ir tik po to krėsti betoną. Tos formuluotės tikslas — projektą sudarančių darbų planavimas. Analizuojant reikia žinoti kiekvieno darbo kainą ir jo pradžios bei pabaigos laiką. Sprendžiant tiesinio programavimo metodais, galima pasirinkti pigiausią schemą, atsižvelgiant į pageidaujamą ir leidžiamą projekto įvykdymo terminą.

## **16. Skaičiavimai ir konstravimas**

Su skaičiavimais ir konstravimu (pavyzdžiui, statybinų konstrukcijų apskaičiavimais) susijusių uždavinių klasę irgi galima traktuoti tiesinio programavimo požiūriu. Tačiau šie klausimai yra pernelyg specialūs, todėl skaitytojui, kuris jais domisi, patariame skaityti bibliografijoje nurodytą literatūrą.

## **17. Dviejų partnerių nulinės sumos lošimas**

Bet kurį dviejų partnerių nulinės sumos lošimą galima išreikšti tiesinio programavimo uždaviniu, o tiesinio programavimo uždavinį irgi galima pakeisti dviejų partnerių nulinės sumos lošimu. Kalbant apie lošimą, formuluotė, duali tiesinio programavimo formuluotei pirmojo lošėjo atžvilgiu, yra tiesinio programavimo formuluotė antrojo lošėjo atžvilgiu. Vienos iš tų sričių skaičiavimo metodus galima taikyti, sprendžiant kitos srities uždavinius.

## LITERATURA

### Tiesinis programavimas

1. **Ackoff R. L.** The Development of Operations Research as a Science.—„J. Oper. Res. Soc. Am.“, 1956, vol. 4, No 3.
2. **Allen R. G. D.** Mathematical Economics. N. Y., Macmillan, 1956.
3. **Bellmore M., Nemhauser G. L.** The Travelling Salesman Problem: A Survey.—„Oper. Res.“, 1968, vol. 16, No 3.
4. **Charnes A., Cooper W. W.** Management Models and Industrial Application of Linear Programming. N. Y., Wiley, 1960.
5. **Charnes A., Cooper W. W.** The Stepping Stone Method of Explaining Linear Calculations in Transportation Problems.—„Manag. Sci.“, 1954—1955, vol. 1.
6. **Charnes A., Cooper W. W., Henderson A.** Introduction to Linear Programming. N. Y., Wiley, 1953.
7. **Charnes A., Cooper W. W., Mellon B.** Blending Aviation Gasolines. A Study in Programming Interdependent Activities in an Integrated Oil Company.—„Econometrica“, 1952, vol. 20.
8. **Churchman C. W., Ackoff R. L., Arnoff E. L.** Introduction to Operations Research. N. Y., Wiley, 1957.
9. **Cooper W. W., Charnes A.** Linear Programming.—„Sci. Am.“, August, 1954.
10. **Dantzig G. B.** Maximisation of a Linear Function of Variables Subject to Linear Inequalities.—In.: Cowles Commission Monograph. Vol. 13. N. Y., Wiley, 1951.
11. **Dantzig G. B.** Application of the Simplex Method to a Transportation Problem.—In.: Cowles Commission Monograph. Vol. 13. N. Y., Wiley, 1951.
12. **Dantzig G. B.** Linear Programming. Princeton, Princeton Univ. Press, 1963.
13. **Dennis J. B.** A High-speed Computer Technique for the Transportation Problem.—„J. Ass. Comput. Mach.“, 1958, vol. 5, No 2.
14. **Dixon J. C., Frederic F. P.** A Decision Rule for Improved Efficiency in Solving Linear Problems with the Simplex Algorithm.—„Communicat. ACM“, 1960, vol. 3, No 9.
15. **Dorfman R.** Application of Linear Programming to the Theory of the Firm. Berkley, Univ. Calif. Press, 1951.
16. **Dorfman R.** Mathematical, or „Linear“, Programming.—„Am. Econom. Rev.“, 1953, December, vol. 43.
17. **Dorfman R., Samuelson P. A., Solow R.** Linear Programming and Economic Analysis. N. Y., McGraw-Hill, 1958.
18. **Eisemann K., Lourie J. R.** The Machine Loading Problem. N. Y., IBM Applications Library, 1959.
19. **Gale D.** The Theory of Linear Economic Models. N. Y., McGraw-Hill, 1960.
20. **Garvin W. W.** Introduction to Linear Programming. N. Y., McGraw-Hill, 1960.
21. **Gass S. L.** Recent Developments in Linear Programming.—„Advances in Computers“, 1961, vol. 2.

22. Good H. W. The Application of a Highspeed Computer to the Definition and Solution of the Vehicular Traffic Problem.— „J. Oper. Res. Soc.“, 1957, vol. 5, No 6.
23. Hadley G. Linear Programming. Reading, Addison-Wesley, 1962.
24. Hayman J., Prager W. Automatic Minimum Weight Design of Steel Frames.— „J. Franklin Inst.“, 1958, vol. 266, No 5.
25. Kelley J. E., Jr. The Cutting-plane Method for Solving Convex Programs. Ambler, Mauchly Ass., 1959.
26. Kelley J. E., Jr. Parametric Programming and Primal-Dual Algorithm.— „Oper. Res.“, 1959, vol. 7, No 3.
27. Koopmans T. C. (ed.) Activity Analysis of Production and Allocation.— In: Cowles Commission Monograph. Vol. 13. N. Y., Wiley, 1951.
28. Lemke C. E. The Dual Method of Solving the Linear Programming Problem.— „Nav. Res. Logist. Quart.“, 1954, vol. 1, No 1.
29. Lemke C. E., Charnes A., Sienkiewicz O. C. Plastic Limit Analysis and Integral Linear Programs.— „Mathematical Report“, 1959, No 21, Troy, Rensselaer Polytechn. Inst.
30. Luce R. D., Raiffa H. Games and Decisions. N. Y., Wiley, 1957.
31. Мак-Кинси Дж. Введение в теорию игр. М., Физматгиз, 1960.
32. Riley V., Gass S. I. Bibliography on Linear Programming and Related Techniques. Baltimore, Johns Hopkins Press, 1958.
33. Saaty T. L. Mathematics Methods of Operations Research. N. Y., McGraw-Hill, 1959.
34. Tucker A. W. An Integer Program for a Multiple-trip Variant of the Travelling Salesman Problem. Princeton, Princeton Univ., 1960.
35. Vajda S. Mathematical Programming, Reading, Addison-Wesley, 1958.
36. Vajda S. Readings in Mathematical Programming. N. Y., Wiley, 1962.
37. Wagner H. M. A Comparison of the Original and Revised Simplex Methods.— „Oper. Res.“, 1957, vol. 5, No 3.
38. Wagner H. M. A Practical Guide to the Dual Theorem.— „Oper. Res.“, 1958, vol. 6, No 3.
39. Wagner H. M. The Simplex Method for Beginners.— „Oper. Res.“, 1958, vol. 6, No 2.
40. Wagner H. M. The Dual Simplex Algorithm for Bounded Variables.— „Nav. Res. Logist. Quart.“, 1958, vol. 5, No 3.
41. Wolf P. The Composite Simplex Algorithm.— „SIAM Rev.“, 1965, vol. 7, No 1.
42. Гасс С. И. Линейное программирование (методы и приложения). М., Физматгиз, 1961.
43. Грешер М. Стратегические игры. Теория и приложения. М., «Советское радио», 1964.
44. Линейные неравенства. Сб. переводов. М., ИЛ, 1959.
45. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование (теория, методы и приложения). М., «Наука», 1969.
46. Оуэн Г. Теория игр. М., «Мир», 1971.

## Tiesnio programavimo laikymai

### *Metalurgija*

**Fabian T.** Application of Linear Programming to Steel Production Planning.—„J. Oper. Res. Soc. Am.“, 1955, vol. 3, No 4.

**Fabian T.** Blast Furnace Burdening and Production Planning.—„Manag. Sci.“, 1967, vol. 14, No 2.

**Reinfeid N. V.** Do You Want Production or Profit?—„Tool. a. Product.“, 1954, vol. 20, No 5.

### *Anglies pramonė*

**Henderson J. M.** A Short-run Model for the Coal Industry.—„Rev. Econom. a. Statist.“, 1955, vol. 37, No 4.

### *Chemijos pramonė*

**Arnoff E. L.** The Application of Linear Programming.—In: Proceedings of Conference on Operations Research in Production and Inventory Control. Cleveland, Case Inst. Technol., 1954.

**Dantzig G. B., Johnson S., White W.** A Linear Programming Approach to the Chemical Equilibrium Problem.—„Manag. Sci.“, 1958, vol. 5, No 1.

### *Naftos pramonė*

**Aronofsky J. S., Williams A. C.** The Use of Linear Programming and Mathematics Models in Underground Oil Production.—„Manag. Sci.“, 1962, vol. 8, No 4.

**Catchpole A. R.** The Application of Linear Programming to Integrated Supply Problems in the Oil Industry.—„Oper. Res. Quart. (U. K.)“, 1962, vol. 13, No 2.

**Charnes A., Cooper W. W., Mellon R.** Blending Aviation Gasolines. A Study in Programming Interdependent Activities in an Integrated Oil Company.—„Econometrica“, 1952, vol. 20, No 2.

**Faur P.** Elements for Selection of Investments in the Refining Industry.—In: Proceedings of the Second International Conference on Operational Research. N. Y., Wiley, 1960.

**Garvin W. W., Crandell H. W., John J. B. a. o.** Applications of Linear Programming in the Oil Industry.—„Manag. Sci.“, 1957, vol. 3, No 4.

**Manne A. S.** Concave Programming For Gasoline Blends.—„J. Oper. Res. Soc. Am.“, 1953, vol. 1, No 3.

**Manne A. S.** Scheduling of Petroleum Refinery Operations.—„Harvard Economic Studies“, 1956, No 48. Cambridge, Harvard Univ. Press.

**Symonds G. H.** A Crude Allocation Problem.—In: **A. S. Manne.** Linear Programming: The Solution of Refinery Problems. N. Y., Esso Stand. Oil. Comp., 1955.

**Symonds G. H.** Optimum Production Rates and Inventory to Meet Uncertain Seasonal Requirements.—In: **A. S. Manne.** Linear Programming: The Solution of Refinery Problems. N. Y., Esso Stand. Oil Comp., 1955.

**Vazsonyi A.** Optimizing a Function of Additively Separated Variables Subject to a Simple Restriction.—„Proceedings of the Second Symposium in Linear Programming“, 1955, vol. 11.

#### *Popieriaus pramonė*

**Gilmore P. C., Gomory R. E.** A linear-programming Approach to the Cutting-stock Problem. Part I.—„Oper. Res.“, 1961, vol. 9, No 6.

**Gilmore P. C., Gomory R. E.** A Linear-programming Approach to the Cutting-stock Problem. Part II.—„Oper. Res.“, 1963, vol. 11, No 6.

**Morgan J. I.** Survey of Operations Research.—„Paper Mill News“, 1961, vol. 84, No 11.

**Paul A. E.** Linear Programming. A Key to Optimum Newsprint Production.—„Pulp a. Paper Mag. Can.“, 1956, vol. 57, No 1.

#### *Kitos pramonės šakos*

**Androit J., Gaussens J.** Programme of Thermal Reactor and „Breeder Reactor“ Power Stations. Fuel Economy Problem of Storage—Price of Plutonium.—In: Proceedings of the Second International Conference on Operational Research. N. Y., Wiley, 1960.

**Charnes A., Cooper W. W., Ferguson R. O.** Optimal Estimation of Executive Compensation by Linear Programming.—„Manag. Sci.“, 1955, vol. 1, No 2.

**Horowitz J., Lattes R., Parker E.** Some Operational Problems Connected with Power Reactor Discharges.—In: Proceedings of the Second International Conference on Operational Research. N. Y., Wiley, 1960.

#### *Transportas ir ryšiai*

**Charnes A., Miller M. H.** A Model for Optimal Programming of Railway Freight Train Movements.—„Manag. Sci.“, 1956, vol. 3, No 1.

**Crane R. R.** A New Tool: Operations Research.—„Mod. Railroads“, 1954, vol. 9, No 1.

**Morton G.** Application of Linear Programming Methods to Commercial Airline Operations.—„Econometrica“, 1953, vol. 21, No 1.

**Kalaba R. E., Juncosa M. L.** Optical Design and Utilization of Communication Networks.—„Manag. Sci.“, 1956, vol. 3, No 1.

**Saaty T. L., Suzuki G.** A Nonlinear Programming Model in Optimum Communication Satellite Use.—„SIAM Rev.“, 1965, vol. 7, No 3.

#### *Gamybos planavimas ir produkcijos sandėliavimas*

**Bellman R. E.** Mathematical Aspects of Scheduling Theory.—„J. Ind. a. Appl. Math.“, 1956, vol. 4, No 3.

**Cahn A. S., Jr.** Warehouse Problem (Abstract 505).—„Bull. Am. Math. Soc.“, 1948, November, vol. 54.

**Charnes A., Cooper W. W.** Generalizations of the Warehousing Model.—„Oper. Res. Quart.“, 1955, vol. 6, No 4.

**Charnes A., Cooper W. W., Farr D.** Linear Programming and Profit Preference Scheduling for a Manufacturing Firm.—„J. Oper. Res. Soc. Am.“, 1953, vol. 1, No 3.

**Charnes A., Cooper W. W., Mellon B.** A Model for Optimizing Production by Reference to Cost Surrogates.— „Econometrica“, 1955, vol. 23, No 3.

**Efroynison M. A., Ray T. L.** A Branch-bound Algorithm for Plant Location.— „Oper. Res.“, 1966, vol. 14, No 3.

**Fetter R. B.** A Linear Programming Model for Long Range Capacity Planning.— „Manag. Sci.“, 1966, vol. 7, No 4.

**Gomory R. E., Dzielinski B. P.** Optimal Programming of Lot Sizes, Inventory and Labor Allocation.— „Manag. Sci.“, 1965, vol. 11, No 9.

**Gepfert A., Grace Ch. H.** Operations Research... as It Is Applied to Production Problems.— „Tool Eng.“, 1956, vol. 36, No 5.

**Johnson S. M.** Optimal Two-and Three-stage Production Schedules with Setup Times Included.— „Nav. Res. Logist. Quart.“, 1954, vol. 1, No 1.

**Канторович Л. В.** Методы оптимизации и математические модели экономики.— УМН, 1970, № 5, 25.

**Kelley J. E., Jr.** Critical Path Planning and Scheduling: Mathematical Basis.— „Oper. Res.“, 1961, vol. 9, No 3.

**Koenigsberg E.** Some Industrial Applications of Linear Programming.— „Oper. Res. Quart. (U. K.)“, 1961, vol. 19, No 2.

**Magee J. R.** Guides to Inventory Policy. Part I: Functions and Lot Sizes.— „Harv. Bus. Rev.“, 1956, vol. 34, No 1; Part II: Problems of Uncertainty.— Vol. 34, No 2; Part III: Anticipating Future Needs.— Vol. 34, No 3.

**Magee J. R.** Linear Programming in Production Scheduling.— „J. Oper. Res. Soc. Am.“, 1953, vol. 1, No 2.

**Manne A. S.** An Application of Linear Programming to the Procurement of Transport Aircraft.— „Manag. Sci.“, 1956, vol. 2, No 2.

**Rapoport L. A., Drews W. P.** Mathematical Approach to Long-range Planning.— „Harvard Business Rev.“, 1962, vol. 40, No 3.

**Salveson M. E.** The Assembly Line Balancing Problem.— In: Proceedings of the Second Symposium in Linear Programming. Vol. 1, 1955.

**Smith S. B.** Planning Transistor Production by Linear Programming.— „Oper. Res.“, 1965, vol. 13, No 1.

**Vazsonyi A.** A Problem in Machine Shop Loading.— „J. Oper. Res. Soc. Am.“, 1955, vol. 3, No 1.

**Whitin T. M.** Inventory Control Research: A Survey.— „Manag. Sci.“, 1954, vol. 1, No 1.

### *Zemės ūkis*

**Boles J. N.** Linear Programming and Farm Management Analysis.— „J. Farm Econom.“, 1955, vol. 37, No 1.

**Candler W.** A Modified Simplex Solution for Linear Programming with Variable Capital Restriction.— „J. Farm Econom.“, 1956, vol. 38, No 4.

**Fisher W. D., Shruben L. W.** Linear Programming Applied to Feedmixing under Different Price Conditions.— „J. Farm Econom.“, 1953, vol. 35, No 4.

**Fox K. A., Taeuber R. C.** Spatial Equilibrium Models of the Livestock-feed Economy.— „Am. Econom. Rev.“, 1955, vol. 45, No 4.

**Fox K. A., Taeuber R. C.** A Spatial Equilibrium Model of the Livestock-feed Economy in the United States.— „Econometrica“, 1953, vol. 21, No 4.

**Hildreth C. G.** Economic Implications of Some Cotton Fertilizer Experiments.— „Econometrica“, 1955, vol. 23, No 1.

**Hildreth C. G., Reiter S.** On the Choice of a Crop Rotation Plan.— In: Cowles Commission Monograph. Vol. 13. N. Y., Wiley, 1951.

**King R. A.** Use of Economic Models: Some Applications of Activity Analysis in Agricultural Economics.— „J. Farm Econom.“, 1953, vol. 35, No 5.

**Pierson D. R.** Farm Profits up by 40 Percent.— „Automat. Data Process.“, 1962, vol. 4, No 5.

**Swanson E. R.** Integrating Crop and Livestock Activities in Farm Management Activity Analysis.— „J. Farm Econom.“, 1955, vol. 37, No 5.

**Swanson L. W., Woodruff J. G.** A Sequential Approach to the Feedmix Problem.— „Oper. Res.“, 1964, vol. 12, No 1.

### *Ekonominiai tyrimai*

**Altshul E.** Reorientation in Economic Theory: Linear and Nonlinear Programming.— „Am. Math. Month.“, 1955, vol. 62, No 6.

**Beckmann M. J., Marschak T.** An Activity Analysis Approach to Location Theory.— In: Proceedings of the Second Symposium in Linear Programming. Vol. 1, 1955.

**Brown J. A. C.** An Experiment in Demand Analysis: The Computation of the Manchester Computer.— „Oper. Res. (JORSA)“, 1956, vol. 4, No 1.

**Charnes A., Cooper W. W.** An Example of Constrained Games in Industrial Economics.— „Econometrica“, 1954, vol. 22, No 4.

**Chipmann J.** Linear Programming.— „Rev. Econom. a Statist.“, 1953, vol. 35, No 2.

**Davidson D., Suppes P.** Experimental Measurement of Utility by Use of a Linear Programming Model.— „Econometrica“, 1956, vol. 24, No 2.

**Dorfman R.** Application of Linear Programming to the Theory of the Firm, Including an Analysis of Monopolistic Firms by Non-linear Programming.— „Econometrica“, 1954, vol. 22, No 1.

**Gunter P.** Use of Linear Programming in Capital Budgeting.— „J. Oper. Res. Soc. Am.“, 1955, vol. 3, No 2.

**Markowitz H. M.** Portfolio Selection.— „J. Finance“, 1952, vol. 7, No 1.

**Martin A. D., Jr.** Mathematical Programming of Portfolio Selections.— „Manag. Sci.“, 1955, vol. 1, No 2.

**Samuelson P. A.** Linear Programming and Economic Theory.— In: Proceedings of the Second Symposium in Linear Programming. Vol 1, 1955.

**Whitin Th. M.** Classical Theory, Graham's Theory and Linear Programming in International Trade.— „Quart. J. Econom.“, 1953, vol. 67, No 4.

### *Skaitiavimai ir konstravimas*

**Charnes A., Greenberg H. J.** Plastic Collapse and Linear Programming. Preliminary Report.— „Bull. Soc.“, 1951, vol. 57, No 6.

**Dorn W. S., Greenberg H. J.** Linear Programming and Plastic Limit Analysis of Structures.— „Technical Report“, 1955, No 7, Pittsburgh, Carnegie Inst. Techn.

**Heyman J.** Plastic Design of Beams and Plane Frames for Minimum Material Consumption.— „Quart. Appl. Math.“, 1951, vol. 8, No 4.

### *Eismo reguliavimas*

**Lavalley R. S.** The Application of Linear Programming to the Problem of Scheduling Traffic Signals.— „J. Oper. Res. Soc. Am.“, 1955, vol. 3, No 4.

**Little J. P. C.** The Synchronization of Traffic Signals by Mixedinteger Linear Programming.— „Oper. Res.“, 1966, vol. 14, No 4.

### *Kiti pritaikymai*

**Cohen K. J., Hammer F. S.** Optimal Coupon Schedules for Municipal Bonds.— „Manag. Sci.“, 1965, vol. 12, No 1.

**Ford L., Fulkerson D. R.** Maximal Flow through a Network.— „Can. J. Math.“, 1956, vol. 8, No 3.

**Freeman R. J., Gogerty D. C., Graves G. W. a. o.** A Mathematical Model of Supply Support for Space Operations.— „Oper. Res.“, 1966, vol. 14, No 1.

**Hartung P. H.** Brand Switching and Mathematical Programming in Market Expansion.— „Manag. Sci.“, 1965, vol. 11, No 10.

**Knight U. G. W.** The Logical Design of Electrical Networks Using Linear Programming Methods.— „Inst. Electr. Eng. Proceed.“, 1960, vol. 107, No 33.

**Kolesar P. J.** Linear Programming and the Reliability of Multi-component Systems.— „Nav. Res. Logist. Quart.“, 1967, vol. 14, No 3.

**Loucks D. P., Revelle C. S., Lynn W. R.** Linear Programming Models for Water Pollution Control.— „Manag. Sci.“, 1967, December, vol. 4.

**McGuire C. B.** Some Team Models of a Sales Organization.— „Manag. Sci.“, 1961, vol. 7, No 2.

**Mannos M.** An Application of Linear Programming to Efficiency in Operations of a System of Dams.— „Econometrica“, 1955, vol. 23, No 3.

**Orden A.** Application of Linear Programming to Optical Filter Design.— In: Proceedings of the Second Symposium in Linear Programming. Vol. 1, 1955.

**Wagner H. M., Giglio R. S., Glaser R. G.** Preventive Maintenance Scheduling by Mathematical Programming.— „Manag. Sci.“, 1964, vol. 10, No 2.

**Wardle P. A.** Forest Management and Operational Research: A Linear Programming Study.— „Manag. Sci.“, 1965, vol. 11, No 10.

**Wilson R. C.** A Packaging Problem.— „Manag. Sci.“, 1965, vol. 12, No 2.

**Percus J., Quinto L.** The Application of Linear Programming to Competitive Bond Bidding.— „Econometrica“, 1956, vol 24, No 4.

**Stanley E. D., Honig D., Gainen L.** Linear Programming in Bid Evaluation.— „Nav. Res. Logist. Quart.“, 1954, vol. 1, No 1.

**Waggener H. A., Suzuki G.** Bid Evaluation for Procurement of Aviation Fuel at DFSC: A Case History.— „Nav. Res. Logist. Quart.“, 1967, vol. 14, No 1.

**Jacobs W. W.** The Caterer Problem.— „Nav. Res. Logist. Quart.“, 1953, vol. 1, No 2.

**Jacobs W. W.** Military Applications of Linear Programming.— In: Proceedings of the Second Symposium in Linear Programming. Vol. 1, 1955.

**Nicholson G. E., Blackwell G. W.** Game Theory and Defence against Community Disaster.— „Res. Prev.“, 1954, vol. 2, No 3.

**Saaty T. L., Webb K. W.** Sensitivity and Renewals in Scheduling Aircraft Overhaul.— In: Proceedings of the Second International Conference on Operational Research. N. Y., Wiley, 1960.

**Wood M. K., Geisler M. A.** Development of Dynamic Models for Program Planning.— In: Activity Analysis of Production and Allocation (Koopmans, ed.). N. Y., Wiley, 1951.

### *Transporto uždavinys ir tinklų teorija*

**Balinski M. L., Quandt R. E.** On an Integer Program for a Delivery Problem.— „Oper. Res.“, 1964, vol. 12, No 2.

**Dantzig G. B., Johnson D. L.** Maximum Payloads per Unit Time Delivered through an Air Network.— „Oper. Res.“, 1964, vol. 12, No 2.

**Dantzig G. B., Ramser J. H.** The Truck Dispatching Problem.— „Manag. Sci.“, 1959, vol. 6, No 1.

**Dwyer P. S.** The Solution of the Hitchcock Transportation Problem with a Method of Reduced Matrices.— In: Ann. Arbor, Univ. Michigan, 1955.

**Flood M. M.** Application of Transportation Theory to Scheduling a Military Tanker Fleet.— „J. Oper. Res. Soc. Am.“, 1954, vol. 2, No 2.

**Flood M. M.** On the Hitchcock Distribution Problem.— „Pacific J. Math.“, 1953, vol. 3, No 2.

**Fulkerson D. R., Dantzig G. B.** Computation of Maximal Flows in Networks.— „Nav. Res. Logist. Quart.“, 1955, vol. 2, No 4.

**Gleyzel A. N.** An Algorithm for Solving the Transportation Problem (Research Paper 2583).— „J. Res. Nat. Bureau Stand.“, 1955, vol. 54, No 4.

**Hitchcock F. L.** The Distribution of a Product from Several Sources to Numerous Localities.— „J. Math. Physics“, 1941, vol. 20, No 3.

**Канторович Л. В.** О перемещении масс.— ДАН СССР, 1942, т. 37, № 7—8.

**Koopmans T. C.** Optimum Utilization of the Transportation System.— „Econometrica“, 1949, vol. 17, No 3 and 4.

**Leiderman J., Gleiberman L., Egan J. F.** Vessel Allocation by Linear Programming.— „Nav. Res. Logist. Quart.“, 1966, vol. 13, No 3.

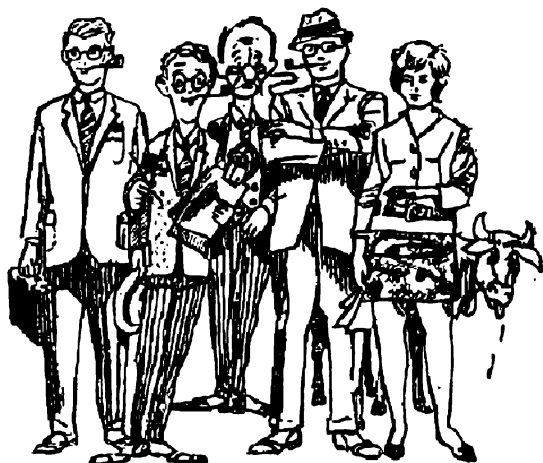
**Orden A.** The Transshipment Problem.— „Manag. Sci.“, 1956, vol. 2, No 2.

**Dantzig G. B., Fulkerson D. R., Johnson S.** Solution of a Large-scale Travelling-salesman Problem.— „J. Oper. Res. Soc. Am.“, 1954, vol. 2, No 4.

**Flood M. N.** The Travelling-salesman Problem.— „Oper. Res.“, 1956, vol. 4, No 1.

**Heller I.** On the Travelling Salesman's Problem.— In: Proceedings of the Second Symposium in Linear Programming. Vol. II, 1955.

**Little J. D. C., Murty K. G., Sweeney D. W. a. o.** An Algorithm for the Travelling-salesman Problem.— „Oper Res.“, 1963, vol. 11, No 6.



## **TURINYS**

|                                                              |     |
|--------------------------------------------------------------|-----|
| Rusiškojo leidimo pratarinė .....                            | 3   |
| Pratarinė .....                                              | 9   |
| 1. Įvadas .....                                              | 10  |
| 2. Uždavinių formulavimas .....                              | 30  |
| 3. Uždavinių sprendimas .....                                | 62  |
| 4. Tiesinis popuri .....                                     | 80  |
| Matematinis priedas ir tiesinio programavimo taikymų sąrašas | 115 |
| Literatūra .....                                             | 147 |

**Gasas S.**

Ga424 Kelionė į tiesinio programavimo šalį. V.,  
„Mokslas“, 1977 ©

156 p. su iliustr. (Matematikos m-kla. 14.) Lit. są-  
rašas: p. 147—155.

Knygoje aiškinama, kodėl įvairiausių sričių specialistai nėra išsiversti be matematinių optimalaus valdymo metodų, kaip programavimo, parodoma, kaip grynai praktinį uždavinį pateiktą jo matematiniu modeliu, nušviečiamas to modelio ir realiosios tikrovės santykis. Skiriama plačiam skaitytojų ratul.

G 20204—215  
M854(10)—77—51—77

47.8

Саял Гасс

ПУТЕШЕСТВИЕ В СТРАНУ  
ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

На литовском языке

Перевел с русского издания 1973 г. Пятрас Румшас

ЛитССР, Вильнюс, издательство «Мокслас», 1977

Saulas Gasas

KELIONĖ Į TIESINIO PROGRAMAVIMO ŠALĮ

Vertė P. Rumšas

Redaktorė E. Petrėnaitė

Viršelio dailininkas A. Kazakauskas

Meninis redaktorius A. Zvilius

Techninė redaktorė N. Marozaitė

Korektorės: N. Žadavičiūtė, G. Zaščinskienė

IB Nr. 196

Duota rinkti 1977.VI.24. Pasirašyta spaudai 1977.XI.17.  
Popierius spaudos Nr. 1. Formatas 84×108<sup>1</sup>/<sub>32</sub>. 8,19 sp. l., 8,71 apsk. l. l.  
Tiražas 6000 egz. Kaina 45 kp

Leidykla „Mokslas“, Vilnius, Stierakausko g. 15.

Spausdino „Pergalės“ spaustuvė, Vilnius, Latako g. 6.  
Užsak. Nr. 2351.